WDS-JniPMML-XLL Documentation

Copyright (c) 2019, Wypasek Data Science, Inc.

General ReadMe

The primary objective of WDS-JniPMML-XLL is to provide model evaluators to Excel. In particular, access to the standard PMML evaluator is a starting point, both for use or for comparison. Later versions will be include other model specs and implement other evaluators.

See documentation articles for a brief introduction.

Through this version, WDS-JniPMML-XLL provides:

- A pair of Excel AddIns (XLLs) and VBA support for:
  - Evaluating PMML models
    - As an Excel function call
    - Using the *de facto* standard implementations
    - Using input data from an in-worksheet table
      Uses XmlMap'd exportable ListObjects, but provides tools to facilitate
    - Can evaluate one or multiple observations (rows) per call
    - Results returned as normal function outputs
    - With cachable models for efficiency

  - Additional data wrangling tools for
    - Importing/Exporting HDF5 compound datasets
    - Importing/Exporting flat files

  - Additional VBA module handling

- A Java wrapper of jpmml.evaluator
  - Callable from the XLL via jni
  - Testable as a standalone from the command line
    But, can be called through the Excel AddIn using the JVM.
  - Input and output data can be:
    - HDF5 compound datasets
    - Flat files
    - In memory (as when called through jni)

- Examples are included
  - A test workbook and launch .bat to run the AddIns without installing
  - A test set of the usual PMML cases

Prerequisites

- 64 bit Excel
  Although, if compiling, 32 bit could possibly be added.
- Access to the VBA project object model (if using the VBA module handlers
- HDF5 and HDFView
  - The HDF5 and HDFView libs are required if compiling, but the functionality could be removed.
  - The provided jars require at least HDFView be on the path or the path passed in as a command line option when starting Excel

- Java jdk-12
  Required when using the latest HDFView install.
- Compiling environment
  The github configurations are for Visual Studio Community Edition and Intellij Community edition.
- DocFx

DocFx is use for the documentation build, including the DocFxDoclet for on the JavaDoc side.

License Note

All code contributions and development from Wypasek Data Science, Inc. (WDataSci) published on its public github site is released under the MIT license. Code from other sources is noted as such, and any assemblies, XLL's, and/or jars that may contain other software (for example, as Apache's Maven or ExcelDna may bundle from other sources) are released along with the commonly used IDE project and/or solution files used to generate them.

TODOs, version 0.5.0

Outstanding items and items for the next version:

- Additional documentation and expanded test suite.
- Date and DateTime datatypes are not fully implemented. They are preliminarily setup to pass as doubles, but wrangling of string values, detection via cell information, and testing needs to be performed.
- The object cache and Handle/Tag handlers could be written better.
- Additional HDF5 utilities, such as returning a layout and querying an element when an HDF5 is used like a memory mapped file.
- Expanded VBA library.

An interesting thought or wish list:

- External memory mapped files, using the DBB wrangler.
- A parallel assembly for calling from MSSQLServer.

Related projects to be published on WDataSci's github site:

- WDataSci XML Model Specification, documentation and tools, including xsl implementation transformations.
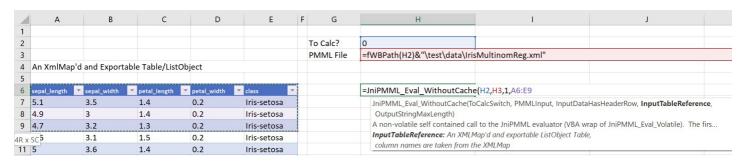- WDataSci Systems Model, documentation and tools.

Articles

Brief Intro

The primary objective of WDS-JniPMML-XLL is to provide model evaluators to Excel. In particular, access to the standard PMML evaluators is a starting point, both for use or for comparison. Later versions will be include other model specs and implement other evaluators.

A quick easy way to evaluate PMML should be available to anyone, even those without access to the latest data science tools. In the finance industry, Excel is ubiquitous. To academic-type data scientists who might scoff at using Excel for anything, it is still a tool which can be used smartly or extremely poorly (which can certainly be said of Python, R, Java, C#, or anything else).

Simple Example

For one-time evaluations (less efficient, but simple), data is arranged in a table object. A macro is provided to assist in providing a technical requirement on the table. The JniPMML_Eval_WithoutCache function takes just a few arguments as below:

In older versions of Excel, results could be returned as an array-valued function, but in Excel 2016, the *spill* feature allows the function to return a dynamic number of rows and columns:

One thing that might not be obvious from the images above is these are function results. The inputs may even be randomized and (relatively) instantaneous evaluations returned.

Slightly More Complex

More efficient model evaluation involves caching the model and then repeated calls to the evaluator without having to do all of the parsing process of the model implementation spec for every calculation. There are several steps involved, but are simplified in the workbook:

- Pick a *Tag* for model
  If you are thinking this should be called a *Handle*, that would seem correct. Except, the Handles are provided and controlled by the Java side and Tags are used on the Excel/C# side.

  A Handle actually has two parts, *HandleMajor.HandleMinor*. The HandleMajor is unique to the Tag and the cached model on the Java side. The HandleMinor increments with subsequant configuraton changes, such as caching the input schema from the ListObject and the output schema. Why do this? It is an Excel-trick. When other ranges depend on the HandleMajor.HandleMinor value of some cell, and that cell recalculates, the correct cascade of recalculations occurs.

- Provide a PMML Model

Here the model can be either a path to a PMML file or (more interestingly) a full PMML file as a string. Why take the string? One could build the file in the workbook. Perhaps one is testing some transformation structure or just wants to see what happens.

- Create a handle
  Cache the PMML model on the Java side and return the new handle to Excel.

- Cache the input and output headers
  Based on the input XmlMap'd exportable table, update the HandleMinor. This also internally caches the model outputs which can be queried for column headings.
  Note: Some PMML models are harder to fully determine the output structures and results are returned as a dictionary-like structure. In the current version and for this case, there is a function, JniPMML_Expand_ComplexValue that can be used to return an expansion. See the WDS-JniPMML-XLL-Test.xlsm workbook for an example.

- Point to input and return the evaluation
  The input is an XmlMap'd exportable table. Before that raises any concerns, there is a macro available through the ribbon, "Add XmlMap to Selected ListObject", which will assign one to it through the following steps:
  - Select a cell in a table or the entire table
  - Hit the macro and you will be queried for one of the following
    - Point to an external XSD file
    - Point to an XSD as a string in a cell
    - Infer one from the table
    - Use a cached PMML dictionary, matching by column name, and infer where a column is not in the dictionary

The slightly more complex example:

	A	B	C	D	E	F	G	H	I
1							Tag	IrisMultinom	
2							PMML File	=fWBPath(H1)&"\test\data\IrisMultinomReg.xml"	
3							Handle	=JniPMML_CreateHandle("JniPMML",H1,H2)	
4	An XmlMap'd and Exportable Table/ListObject						Cache I/O Headers	=JniPMML_Eval_CacheHeaders(H3,A6)	
5									
6	sepal_length	sepal_width	petal_length	petal_width	class		Return Header	=JniPMML_Eval_OutputColumnHeadings(H4)	
7	5.1	3.5	1.4	0.2	Iris-setosa		Single Row	=JniPMML_Eval(H4,0,A7:E7)	
8	4.9	3	1.4	0.2	Iris-setosa		Multiple Rows	=JniPMML_Eval(H4,0,A8:E10)	
9	4.7	3.2	1.3	0.2	Iris-setosa			JniPMML_Eval(HandleOrTag, InputDataIncludesHeader, **InputData**)	
10	4.6	3.1	1.5	0.2	Iris-setosa			Calls JniPMML.Eval based a previously set Header	
11	5	3.6	1.4	0.2	Iris-setosa			***InputData:*** *Select Contiguous ListObject Rows, include header if needed for alignment.*	

With result:

	A	B	C	D	E	F	G	H	I	J	K
1							Tag	IrisMultinom			
2							PMML File		\WDS-JniPMML-XLL\trunk\test\data\IrisMultinomReg.xml		
3							Handle	1.1			
4	An XmlMap'd and Exportable Table/ListObject						Cache I/O Headers	1.3			
5											
6	sepal_leng	sepal_widt	petal_leng	petal_widt	class		Return Header	class-predictedValue	Probability_Iris-setosa	Probability_Iris-versicolor	Probability_Iris-virginica
7	5.1	3.5	1.4	0.2	Iris-setosa		Single Row	Iris-setosa	1	3.08813E-23	0
8	4.9	3	1.4	0.2	Iris-setosa		Multiple Rows	Iris-setosa	1	7.36829E-15	0
9	4.7	3.2	1.3	0.2	Iris-setosa			Iris-setosa	1	4.46658E-17	0
10	4.6	3.1	1.5	0.2	Iris-setosa			Iris-setosa	1	7.84035E-12	0

See the provided Excel test workbook for additional examples.

Additional Usage Notes

- Accessing Java via JNICode creates a COM AddIn
  Efforts have been made to make sure COM objects are clean up. However, should the process break for whatever reason, there may be an Excel process hanging around. In that case, look in the taskmgr's details. Or, use something like the powershell snippets in the scripts folder to find and stop.

Notes on JniPMML

Author: Christian Wypasek

Simple Motivation

My daughter, a college student, asked me to explain this project in one sentence and this was as close as I could get: Scientists build models. For even something as simple as linear regression, there is a formula that needs to evaluated. It might be for my own purposes, or it might be for a company I work for, but model implementation needs to be easily accessible. Even though data scientists might use special tools, everyone in financial services at least has Excel.

Slightly More Technical Motivation

Regardless of whether or not Excel might be highly regarded as a computational framework among academicians, it is ubiquitous in financial services (even if it might not be used well). Therefore, it makes sense that invoking an XML based evaluator from within Excel would be worthwhile. In particular, since Excel can enable rapid visualization, one should also be able to compare evaluator implementations and view model response to variable changes and/or model structure in a live manner.

XML/PMML

For someone like myself who works across the spectrum of big data projects (project management and business interface, data science, and data engineer) and works across multiple programming languages, consistency of treatments is a fundamental key to efficiency. After years of engineering databases, building complex statistical models for financial instruments, and incorporating these models into asset backed cash flow valuations, the greatest risks in this data science process are often operational. There is the most obvious question, "Is the data being used for forecasting sufficiently like the data the model was fit on?", but one also has to ask "Is the model being calculated correctly?".

From personal experience, hand coding something like a scoring model requires significant quality checks and carries the persistent risk that something was overlooked. It does not take too many hand coding events to make one believe there has got to be a better way, both for efficiency of process and the reduction of mistakes that come from mind numbing exercises. Starting back in 1998/1999, I started using markup styles to facilitate both the modeling process and facilitating the implementation for scoring and other types of regresson and non-parametric models. Since then, PMML (predictive modeling markup language) as become an industry standard.

The PMML standard has evolved and early versions were not sophisticated enough for my needs. For example, the Scorecard implementation was not added until the end of 2011, and transformations were not added until 2014. For all that it is, PMML is still a communication standard for model implementation and is often generated after a model has been fit. Continued diligence is required so the communicated model truly represents the intended relationship between the input data and the output results. A process oriented view of statistical model building starts with data preparation and can be exploited at every step of the process through to final implementation.

There may be more than one way to skin a cat, but very few which leave you with anything looks like a cat. My personal work has included using mathematical and statistical model specifications in XML with implementations in SAS, C++, Python, R, in-database (Vertica, MSSQL) UDTFs in C++/Java/R, and VBA (in Excel). After drilling into PMML implementation details, there is still much to be desired. An updated XML specification used by WDataSci for model fitting and alternate implementations will be released on the its github as a later project, but transformation (such as through XSLT) into PMML for delivery is reasonable given the industry standardization that PMML offers. Other model implementation specifications, such as pfa, will emerge, and Excel will remain a platform for either a model delivery or easy comparison.

WDS-JniPMML as a multi-language project

The JniPMML project combines several APIs, each for a specific purpose:

- Java

  The *de facto* implementation of PMML is jpmml.evaluator. JniPMML-Java wraps the implementation in a manner that creates a standalone jar that can also be called from C# via jni.

- C#

  Using the ExcelDna project to facilitate Excel functionality, the JniPMML-Cs assembly wraps the jni calls.

- VB

  Some odds and ends which I have traditionally done in VBA, but using ExcelDna .Net. In particular, wrangling of the VBA modules is done in VB.

- VBA

  Certain Excel functions created with ExcelDna through either C# or VB become *volatile* in that they recalculate at every calculation event (which can be a bad thing). However, good old fashioned VBA can do the same thing in a non-volatile manner.

Working in different languages for different aspects of a larger project is not unusual. For example, database work might be done in SQL, with processing either in database or written-out-processed-read-back-in, and final summaries might have an entirely different framework. When sub-projects have many parallel functions, the tendency of programmers to have a project on one side and then start from scratch on the other side, can lead to unexpected differences which the programmer then might struggle to balance. Complete one side, move to the other, discover some new or useful treatment, go back to the first side, restart loop. This project also started in that manner.

Passing data back and forth in-memory between Java and C# involves packing memory in a particular way, which also turned out to be the HDF5.PInvoke bulk writes a HDF5 compound dataset (such as R can export). Development of the project included consideration of in-memory HDF5s, which despite HDF5 docs, is not ready for prime-time. For testing purposes, HDF5 CompoundDS and flat file functionality is included in the JniPMML-Java project and the Excel AddIn.

Finally, the Excel AddIn also includes other tools representative of some extended functionality I have come to expect over the years, such as VBA component wrangling and other examples. Even if this project is not used extensively outside of WDataSci, this project also become an in-house reference for C#/Java differences and quirks, DocFx, Excel AddIns (quirks across C#, ExcelDna, VB, VBA, COM, non-COM), PMML (and jpmml quirks), HDF5 (and quirks across HDF.PInvoke, HDF-Java, HDF-Object), etc.

Notes on Java x Cs

Author: Christian Wypasek

The mirroring of the C# and Java code is meant not to be slick or cute. It is simply because both implementations are reading and writing the same formats. When handing off data in a ByteBuffer between C# and Java, in both directions, the formats must be *exactly* the same. (Note, not going down the AST route. It seems like if you are going to go that route, you should be all in.)

Some syntax differences are too big to bridge, such as in how enums are more flexible in Java than C#. With enums, just the values and methods (extensions in C#) are in common. The source codes will still be organized similarly, but this is also why enums are not otherwise in the files with their naturally associated classes.

Some syntax differences are not marked but obvious:

- Non-method properties, such as length/Length or boolean/Boolean, which are easy enough to fix in IDEs.
- To break String object references in Java where C# does not require it, a simple new_String() function in C# is a pass through and differs only with the "_".
- Method *throws* required in Java but not C# are on separate lines and commented out in C#.
- In switch-case statements on enum values where Java case statements do not require qualified names, there will be two lines one uncommented for Java, the other commented for C#, and visa-versa.

The syntax differences for many common methods amount only to the case of the leading letter, such as with Java's String.toString() vs C#'s String.ToString(). When this leading case issue is on a class method, they can be minimized through C#'s static extension methods, included in a static class, JavaLikeExtensions. Why not just let one letter differences ride, like in length/Length above? One line in one file and one less thing to highlight a difference in vimdiff. Other differences can be eliminated through specially named classes, mimicking names and methods used on the Java side, such as Map, PrintWriter, and ArrayList. Even though broken out in the documentation, on C#, they can all be included in the WDSXJava.cs, along with JavaLikeExtensions.

Syntax differences over lines or blocks are handled in two ways: First, when a one line change is required, a comment leading with //Java or //C# precedes the line. On the Java side, the //C# and subsequent line are collapsed, commenting out the C# syntax. The reverse treatment is used on the C# side.

For example, in C# version:

```
//C#
if ( !base.Equals(arg) ) return false;
//Java if ( !super.Equals(arg) ) return false;
```

And in the Java version:

```
//C# if ( !base.Equals(arg) ) return false;
//Java
if ( !super.Equals(arg) ) return false;
```

For larger blocks, we can exploit the behavior that an open-comment /* jumps over other open comments until the first closing */. Therefore, in the C# version (Note that the Java >>> comment is open):

```
/* C# >>> */
if ( !base.Equals(arg) ) return false;
/* <<< C# */
/* Java >>> *
if ( !super.Equals(arg) ) return false;
/* <<< Java */
```

And in the Java version (Note that the C# >>> comment is open):

```
/* C# >>> *
if ( !base.Equals(arg) ) return false;
/* <<< C# */
/* Java >>> */
if ( !super.Equals(arg) ) return false;
/* <<< Java */
```

There are multiple programming languages used in this project:

Note: This documentation bundle was created using DocFx, which was confusing documentation across APIs. Therefore, a separate PDF has been generated for each.

Java APIs

JniPMML-Java

The initial design of JniPMML-Java is to wrap jpmml into a single jar which can be called from Excel/C# via JNICode. However, it is a standalone that can be used with command line calls. It therefore has wranglers for text and HDF5 files in addition to the ByteBuffers for interacting with C#.

WDS-Java

General utilitities that independent of JniPMML code. To simplify assemblies and jars, this code is included in the larger projects, but is also compiled as a stand alone.

On the java side, there is a separate WDS-00.00.00.jar generated but it is pulled into a shaded jar so that only one WDS-JniPMML-00.00.00.jar needs to be used in practice.

C# APIs

JniPMML-Cs

The Java style com.WDataSci namespaces are specifically for C# code which mirrors the Java modules.

The JNI namespace originated externally, but with a few local completion and extensions.

WDS-Cs

General utilitities that independent of JniPMML code. To simplify assemblies and jars, this code is included in the larger projects, but is also compiled as a stand alone.

The Java style com.WDataSci namespaces are specifically for C# code which mirrors the Java modules.

The C# style namespaces are not specifically mirrored in the Java code.

VB APIs

JniPMML-VB

The JniPMML-VB code is primarily for some additional Excel manipulation functionality. In particular, the wrangling the Excel VBE components. The ExcelDna and Microsoft.Office.Interop.Excel libraries are generally mirrored in both C# and VB, however, ExcelDna UDF functions which take references as objects so that information about the caller can be determined at run-time become automatically volatile. For this reason, there are several function wrappers implemented in VBA which must be either in an another addin, or as a VBA module in the workbook. The JniPMML-VB (and supporting WDS-VB code which is pulled into the assembly) addin facilitates these wrapped functions by providing a wrangler for a WDSJniPMML.bas module.

WDS-VB

General utilitities that independent of JniPMML code. To simplify assemblies and jars, this code is included in the larger projects, but is also compiled as a stand alone.

VBA APIs

WDS-VBA

The WDS-VBA code is a collection of VBA macros that can be included in Excel workbooks and there are C#/VB macros accessibly through the ribbon to wrangle them in and out of workbooks as needed.

There is a necessity for at least the WDSCore macro for adding ExcelDna Intellisense capabilities.

Java APIs

JniPMML-Java

The initial design of JniPMML-Java is to wrap jpmml into a single jar which can be called from Excel/C# via JNICode. However, it is a standalone that can be used with command line calls. It therefore has wranglers for text and HDF5 files in addition to the ByteBuffers for interacting with C#.

WDS-Java

General utilitities that independent of JniPMML code. To simplify assemblies and jars, this code is included in the larger projects, but is also compiled as a stand alone.

On the java side, there is a separate WDS-00.00.00.jar generated but it is pulled into a shaded jar so that only one WDS-JniPMML-00.00.00.jar needs to be used in practice.

JniPMML-Java

The initial design of JniPMML-Java is to wrap jpmml into a single jar which can be called from Excel/C# via JNICode. However, it is a standalone that can be used with command line calls. It therefore has wranglers for text and HDF5 files in addition to the ByteBuffers for interacting with C#.

Namespace com.WDataSci.JniPMML

Classes

Cmd

CmdArgs

DBB

DBB.Default

Default

FieldBaseMD

FieldBaseMD contains the cross-package data field information.

It holds cached information for input and output processing and provides the any simplifications, for example, numeric types that are not int, long, byte, or boolean, are taken as double.

On purpose, the enumerated type names are only 3 characters to make the use explicit and not cause any confusion with XSD, PMML, SQL, or HDF5 types.

On the Java side, because of the *finalized* status of org.dmg.pmml.FieldName, which is used as the column key for input and output columns of the jpmml evaluator, handling of the mapping and cache of org.dmg.pmml.FieldNames is handled by the FieldMD implementation of XDataFieldKeyInterface.

FieldBaseMD.Default

FieldMD

FieldMDEnums

JniPMML

JniPMML is the main bridge between callers (at the moment, either from C# or a Java command line) and a wrap around a PMML document and its evaluator. To allow a caching of multiple documents and input/output maps, JniPMML contains a collection of JniPMMLItems which can be called by handle or the last one used.

The intent is to pass data frame like blocks of data as inputs for evaluation. Because of incomplete interfaces in HDF5, this is done

via a ByteBuffer from C#, but Wranglers for FlatFiles, HDF5, DBB (Direct ByteBuffer) and others are or will be implemented.

JniPMMLItem

JniPMMLItem.__ConfigMatter

JniPMMLItem.__InputMatter

JniPMMLItem.__OutputMatter

JniPMMLItem.__PMMLMatter

RecordSet

RecordSetMD

RecordSetMD.__ModeMatter

RecordSetMD.__SchemaMatter

RecordSetMDEnums

Util

WranglerDBB

WranglerDBB.__DBBMatter

WranglerFlatFile

WranglerHDF5

WranglerHDF5.HDF5DataType

WranglerXSD

Interfaces

FieldMDIKey<T>

Enums

FieldMDEnums.eDTyp

FieldMDEnums.eRTyp

RecordSetMDEnums.eMode

RecordSetMDEnums.eSchemaType

RecordSetMDEnums.eType

Class Cmd

Inheritance

java.lang.Object

Cmd

Namespace:

Assembly: .dll

Syntax

```
public class Cmd
```

Constructors

Cmd()

Declaration

```
public Cmd()
```

Methods

main(String[] argv)

Declaration

```
public static void main(String[] argv)
```

Parameters

TYPE	NAME	DESCRIPTION
	argv	

Exceptions

TYPE	CONDITION
java.lang.Exception	

Class CmdArgs

Inheritance

java.lang.Object

CmdArgs

Namespace:

Assembly: .dll

Syntax

```
public class CmdArgs
```

Constructors

CmdArgs()

Declaration

```
public CmdArgs()
```

Fields

help

Declaration

```
public boolean help
```

Field Value

TYPE	DESCRIPTION
boolean	

verbose

Declaration

```
public boolean verbose
```

Field Value

TYPE	DESCRIPTION
boolean	

aBaseDir

Declaration

```
public String aBaseDir
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aPMMLFileName

Declaration

```
public String aPMMLFileName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aInputFileType

Declaration

```
public String aInputFileType
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aInputFileName

Declaration

```
public String aInputFileName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aInputSchemaFileName

Declaration

```
public String aInputSchemaFileName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aInputSchemaRecordSetName

Declaration

```
public String aInputSchemaRecordSetName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aInputSchemaType

Declaration

```
public String aInputSchemaType
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

bCheckForHeaderRow

Declaration

```
public boolean bCheckForHeaderRow
```

Field Value

TYPE	DESCRIPTION
boolean	

bInputHasHeaderRow

Declaration

```
public boolean bInputHasHeaderRow
```

Field Value

TYPE	DESCRIPTION
boolean	

aInputFileDlm

Declaration

```
public String aInputFileDlm
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aOutputFileName

Declaration

```
public String aOutputFileName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aOutputFileType

Declaration

```
public String aOutputFileType
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aOutputHDF5FixedStringLength

Declaration

```
public int aOutputHDF5FixedStringLength
```

Field Value

TYPE	DESCRIPTION
int	

bOutputHeaderRow

Declaration

```
public boolean bOutputHeaderRow
```

Field Value

TYPE	DESCRIPTION
boolean	

aOutputHDF5DataSetName

Declaration

```
public String aOutputHDF5DataSetName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

bOutputInputFields

Declaration

```
public boolean bOutputInputFields
```

Field Value

TYPE	DESCRIPTION
boolean	

aOutputInputFieldNameSuffix

Declaration

```
public String aOutputInputFieldNameSuffix
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aOutputCompositeFieldDlm

Declaration

```
public String aOutputCompositeFieldDlm
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

aOutputFileDlm

Declaration

```
public String aOutputFileDlm
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

Methods

mProcessBaseDir()

Declaration

```
public void mProcessBaseDir()
```

mRecapParameters()

Declaration

```
public String mRecapParameters()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

mRecap()

Declaration

```
public void mRecap()
```

Class DBB

Syntax

```
public class DBB
```

Constructors

DBB(ByteBuffer arg)

Declaration

```
public DBB(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

DBB(byte[] arg)

Declaration

```
public DBB(byte[] arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

DBB(WDataSci.JniPMML.DBB arg, boolean bJustData)

Declaration

```
public DBB(WDataSci.JniPMML.DBB arg, boolean bJustData)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	
boolean	bJustData	

DBB(byte[] arg, int offset, boolean bIsBigEndian)

Declaration

```
public DBB(byte[] arg, int offset, boolean bIsBigEndian)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	
int	offset	
boolean	bIsBigEndian	

DBB(byte[] arg, int offset, int length, boolean bIsBigEndian)

Declaration

```
public DBB(byte[] arg, int offset, int length, boolean bIsBigEndian)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	
int	offset	
int	length	
boolean	bIsBigEndian	

Fields

data

Declaration

```
public byte[] data
```

Field Value

TYPE	DESCRIPTION

datawrap

Declaration

```
public ByteBuffer datawrap
```

Field Value

TYPE	DESCRIPTION
java.nio.ByteBuffer	

bUsingByteBufferOnly

Declaration

```
boolean bUsingByteBufferOnly
```

Field Value

TYPE	DESCRIPTION
boolean	

bHasLeaders

Declaration

```
public boolean bHasLeaders
```

Field Value

TYPE	DESCRIPTION
boolean	

bHasFLenVLenSplit

Declaration

```
public boolean bHasFLenVLenSplit
```

Field Value

TYPE	DESCRIPTION
boolean	

bIsReadOnly

Declaration

```
public boolean bIsReadOnly
```

Field Value

TYPE	DESCRIPTION
boolean	

offset

Declaration

```
public long offset
```

Field Value

TYPE	DESCRIPTION
long	

Length

Declaration

```
public long Length
```

Field Value

TYPE	DESCRIPTION
long	

flenoffset

Declaration

```
public long flenoffset
```

Field Value

TYPE	DESCRIPTION
long	

flenlength

Declaration

```
public long flenlength
```

Field Value

TYPE	DESCRIPTION
long	

vlenoffset

Declaration

```
public long vlenoffset
```

Field Value

TYPE	DESCRIPTION
long	

vlenlength

Declaration

```
public long vlenlength
```

Field Value

TYPE	DESCRIPTION
long	

LayoutStyle

Declaration

```
public String LayoutStyle
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

nDBBRequiredBytes

Declaration

```
public long nDBBRequiredBytes
```

Field Value

TYPE	DESCRIPTION
long	

nDBBLeadingBytes

Declaration

```
public long nDBBLeadingBytes
```

Field Value

TYPE	DESCRIPTION
long	

nDBBFLenBytes

Declaration

```
public long nDBBFLenBytes
```

Field Value

TYPE	DESCRIPTION
long	

nDBBVLenBytes

Declaration

```
public long nDBBVLenBytes
```

Field Value

TYPE	DESCRIPTION
long	

nRecords

Declaration

```
public long nRecords
```

Field Value

TYPE	DESCRIPTION
long	

nRecordFLenBytes

Declaration

```
public long nRecordFLenBytes
```

Field Value

TYPE	DESCRIPTION
long	

nRecordVLenBytes

Declaration

```
public long nRecordVLenBytes
```

Field Value

TYPE	DESCRIPTION
long	

bIsBigEndian

Declaration

```
public boolean bIsBigEndian
```

Field Value

TYPE	DESCRIPTION
boolean	

ptr

Declaration

```
public long ptr
```

Field Value

TYPE	DESCRIPTION
long	

flenptr

Declaration

```
public long flenptr
```

Field Value

TYPE	DESCRIPTION
long	

vlenptr

Declaration

```
public long vlenptr
```

Field Value

TYPE	DESCRIPTION
long	

Methods

Dispose()

Declaration

```
public void Dispose()
```

isValid()

Declaration

```
public boolean isValid()
```

Returns

TYPE	DESCRIPTION
boolean	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

position(long ptr, long flenptr, long vlenptr)

Declaration

```
public void position(long ptr, long flenptr, long vlenptr)
```

Parameters

TYPE	NAME	DESCRIPTION
long	ptr	
long	flenptr	
long	vlenptr	

Reset()

Declaration

```
public void Reset()
```

Wrap(byte[] arg)

Declaration

```
public WDataSci.JniPMML.DBB Wrap(byte[] arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Returns

TYPE	DESCRIPTION

cReadExistingLayout()

Declaration

```
public WDataSci.JniPMML.DBB cReadExistingLayout()
```

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cWrap(ByteBuffer arg)

Declaration

```
public WDataSci.JniPMML.DBB cWrap(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

cWrap(ByteBuffer arg, int offset, boolean bIsBigEndian)

Declaration

```
public WDataSci.JniPMML.DBB cWrap(ByteBuffer arg, int offset, boolean bIsBigEndian)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
int	offset	
boolean	bIsBigEndian	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

cWrap(byte[] arg, int offset, boolean bIsBigEndian)

Declaration

```
public WDataSci.JniPMML.DBB cWrap(byte[] arg, int offset, boolean bIsBigEndian)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	
int	offset	
boolean	bIsBigEndian	

Returns

TYPE	DESCRIPTION

cWrap(byte[] arg, int offset, int length, boolean bIsBigEndian)

Declaration

```
public WDataSci.JniPMML.DBB cWrap(byte[] arg, int offset, int length, boolean bIsBigEndian)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	
int	offset	
int	length	
boolean	bIsBigEndian	

Returns

TYPE	DESCRIPTION

cAsReadOnly()

Declaration

```
public WDataSci.JniPMML.DBB cAsReadOnly()
```

Returns

TYPE	DESCRIPTION

cWithOffset(int offset)

Declaration

```
public WDataSci.JniPMML.DBB cWithOffset(int offset)
```

Parameters

TYPE	NAME	DESCRIPTION
int	offset	

Returns

TYPE	DESCRIPTION

cWithLength(int length)

Declaration

```
public WDataSci.JniPMML.DBB cWithLength(int length)
```

Parameters

TYPE	NAME	DESCRIPTION
int	length	

Returns

TYPE	DESCRIPTION

cAsBigEndian()

Declaration

```
public WDataSci.JniPMML.DBB cAsBigEndian()
```

Returns

TYPE	DESCRIPTION

cAsNotBigEndian()

Declaration

```
public WDataSci.JniPMML.DBB cAsNotBigEndian()
```

Returns

TYPE	DESCRIPTION

cAsSimple()

Declaration

```
public WDataSci.JniPMML.DBB cAsSimple()
```

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

cAsHDF5BulkCompoundDSWriteLayout(long nRecords, long nRecordFLenBytes)

Declaration

```
public WDataSci.JniPMML.DBB cAsHDF5BulkCompoundDSWriteLayout(long nRecords, long nRecordFLenBytes)
```

Parameters

TYPE	NAME	DESCRIPTION
long	nRecords	
long	nRecordFLenBytes	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

cAsUsualLayout(String LayoutStyle, long nLeadingBytes, long nRecords, long nRecordFLenBytes, long nRecordVLenBytes)

Declaration

```
public WDataSci.JniPMML.DBB cAsUsualLayout(String LayoutStyle, long nLeadingBytes, long nRecords, long
nRecordFLenBytes, long nRecordVLenBytes)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	LayoutStyle	
long	nLeadingBytes	
long	nRecords	
long	nRecordFLenBytes	
long	nRecordVLenBytes	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

cAsUsualLayout(String LayoutStyle, long nRecords, long nRecordFLenBytes, long nRecordVLenBytes)

Declaration

```
public WDataSci.JniPMML.DBB cAsUsualLayout(String LayoutStyle, long nRecords, long nRecordFLenBytes, long
nRecordVLenBytes)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	LayoutStyle	
long	nRecords	

TYPE	NAME	DESCRIPTION
long	nRecordFLenBytes	
long	nRecordVLenBytes	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

isDirect()

Declaration

```
public boolean isDirect()
```

Returns

TYPE	DESCRIPTION
boolean	

GetLayerByte(int layer)

Declaration

```
public byte GetLayerByte(int layer)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	

Returns

TYPE	DESCRIPTION
byte	

GetLayerByteAt(int layer, long arg)

Declaration

```
public byte GetLayerByteAt(int layer, long arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	

TYPE	NAME	DESCRIPTION
long	arg	

Returns

TYPE	DESCRIPTION
byte	

GetLayerInt(int layer)

Declaration

```
public int GetLayerInt(int layer)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	

Returns

TYPE	DESCRIPTION
int	

GetLayerInt(int layer, long atarg)

Declaration

```
public int GetLayerInt(int layer, long atarg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	

Returns

TYPE	DESCRIPTION
int	

GetLayerLong(int layer)

Declaration

```
public long GetLayerLong(int layer)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	

Returns

TYPE	DESCRIPTION
long	

GetLayerLong(int layer, long atarg)

Declaration

```
public long GetLayerLong(int layer, long atarg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	

Returns

TYPE	DESCRIPTION
long	

GetLayerDouble(int layer)

Declaration

```
public double GetLayerDouble(int layer)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	

Returns

TYPE	DESCRIPTION
double	

GetLayerDouble(int layer, long atarg)

Declaration

```
public double GetLayerDouble(int layer, long atarg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	

Returns

TYPE	DESCRIPTION
double	

PutLayerInt(int layer, int value)

Declaration

```
public void PutLayerInt(int layer, int value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
int	value	

PutLayerInt(int layer, long atarg, int value)

Declaration

```
public void PutLayerInt(int layer, long atarg, int value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	
int	value	

PutLayerLong(int layer, long value)

Declaration

```
public void PutLayerLong(int layer, long value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	value	

PutLayerLong(int layer, long atarg, long value)

Declaration

```
public void PutLayerLong(int layer, long atarg, long value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	

TYPE	NAME	DESCRIPTION
long	atarg	
long	value	

PutLayerDouble(int layer, double value)

Declaration

```
public void PutLayerDouble(int layer, double value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
double	value	

PutLayerDouble(int layer, long atarg, double value)

Declaration

```
public void PutLayerDouble(int layer, long atarg, double value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	
double	value	

PutLayerBytes(int layer, byte[] value)

Declaration

```
public void PutLayerBytes(int layer, byte[] value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
	value	

PutLayerBytes(int layer, long atarg, byte[] value)

Declaration

```
public void PutLayerBytes(int layer, long atarg, byte[] value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	
	value	

PutLayerZeros(int layer, int value)

Declaration

```
public void PutLayerZeros(int layer, int value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
int	value	

PutLayerZeros(int layer, long atarg, int value)

Declaration

```
public void PutLayerZeros(int layer, long atarg, int value)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	
int	value	

GetLayerFLenString(int layer, long nByteMaxLength)

Declaration

```
public String GetLayerFLenString(int layer, long nByteMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	nByteMaxLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

GetLayerFLenString(int layer, long atarg, long nByteMaxLength)

Declaration

```
public String GetLayerFLenString(int layer, long atarg, long nByteMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	
long	nByteMaxLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

GetLayerVLenString(int layer, long nByteMaxLength)

Declaration

```
public String GetLayerVLenString(int layer, long nByteMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	nByteMaxLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

GetLayerVLenString(int layer, long atarg, long nByteMaxLength)

Declaration

```
public String GetLayerVLenString(int layer, long atarg, long nByteMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	
long	nByteMaxLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

PutLayerFLenString(int layer, String value, int nByteMaxLength, int nZeroBytes)

Declaration

```
public void PutLayerFLenString(int layer, String value, int nByteMaxLength, int nZeroBytes)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
java.lang.String	value	
int	nByteMaxLength	
int	nZeroBytes	

PutLayerFLenString(int layer, long atarg, String value, int nByteMaxLength, int nZeroBytes)

Declaration

```
public void PutLayerFLenString(int layer, long atarg, String value, int nByteMaxLength, int nZeroBytes)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	

TYPE	NAME	DESCRIPTION
java.lang.String	value	
int	nByteMaxLength	
int	nZeroBytes	

PutLayerVLenString(int layer, String value, int nByteMaxLength, int nZeroBytes)

Declaration

```
public void PutLayerVLenString(int layer, String value, int nByteMaxLength, int nZeroBytes)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
java.lang.String	value	
int	nByteMaxLength	
int	nZeroBytes	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

PutLayerVLenString(int layer, long atarg, String value, int nByteMaxLength, int nZeroBytes)

Declaration

```
public void PutLayerVLenString(int layer, long atarg, String value, int nByteMaxLength, int nZeroBytes)
```

Parameters

TYPE	NAME	DESCRIPTION
int	layer	
long	atarg	
java.lang.String	value	
int	nByteMaxLength	
int	nZeroBytes	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

Class DBB.Default

Inheritance

java.lang.Object

DBB.Default

Namespace:

Assembly: .dll

Syntax

```
public static class DBB.Default
```

Constructors

Default()

Declaration

```
public Default()
```

Fields

nLeadingBytes

Declaration

```
public static final long nLeadingBytes
```

Field Value

TYPE	DESCRIPTION
long	

Class Default

Inheritance

java.lang.Object

Default

Namespace:

Assembly: .dll

Syntax

```
public class Default
```

Constructors

Default()

Declaration

```
public Default()
```

Fields

ISJAVA

Declaration

```
public static final boolean ISJAVA
```

Field Value

TYPE	DESCRIPTION
boolean	

ISCSHARP

Declaration

```
public static final boolean ISCSHARP
```

Field Value

TYPE	DESCRIPTION
boolean	

HeaderStringMaxLength

Declaration

```
public static final int HeaderStringMaxLength
```

Field Value

TYPE	DESCRIPTION
int	

StringMaxLength

Declaration

```
public static final int StringMaxLength
```

Field Value

TYPE	DESCRIPTION
int	

anyVLenRead

Declaration

```
public static final Boolean anyVLenRead
```

Field Value

TYPE	DESCRIPTION
java.lang.Boolean	

anyVLenWrite

Declaration

```
public static final Boolean anyVLenWrite
```

Field Value

TYPE	DESCRIPTION
java.lang.Boolean	

Class FieldBaseMD

FieldBaseMD contains the cross-package data field information.

It holds cached information for input and output processing and provides the any simplifications, for example, numeric types that are not int, long, byte, or boolean, are taken as double.

On purpose, the enumerated type names are only 3 characters to make the use explicit and not cause any confusion with XSD, PMML, SQL, or HDF5 types.

On the Java side, because of the *finalized* status of org.dmg.pmml.FieldName, which is used as the column key for input and output columns of the jpmml evaluator, handling of the mapping and cache of org.dmg.pmml.FieldNames is handled by the FieldMD implementation of XDataFieldKeyInterface.

Inheritance

java.lang.Object

FieldBaseMD

Namespace:

Assembly: .dll

Syntax

```
public class FieldBaseMD
```

Constructors

FieldBaseMD()

Declaration

```
public FieldBaseMD()
```

FieldBaseMD(String _Name, WDataSci.JniPMML.FieldMDEnums.eDTyp _DTyp)

Declaration

```
public FieldBaseMD(String _Name, WDataSci.JniPMML.FieldMDEnums.eDTyp _DTyp)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	_Name	
WDataSci.JniPMML.FieldMDEnums.eDTyp	_DTyp	

FieldBaseMD(String Name, WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp, int StringMaxLength)

Declaration

```
public FieldBaseMD(String Name, WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp, int StringMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	Name	
WDataSci.JniPMML.FieldMDEnums.eDTyp	DTyp	

TYPE	NAME	DESCRIPTION
int	StringMaxLength	

FieldBaseMD(WDataSci.JniPMML.FieldBaseMD arg)

Declaration

```
public FieldBaseMD(WDataSci.JniPMML.FieldBaseMD arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

FieldBaseMD(String Name, int hclass, int hlength, int horder, int hsign)

Declaration

```
public FieldBaseMD(String Name, int hclass, int hlength, int horder, int hsign)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	Name	
int	hclass	
int	hlength	
int	horder	
int	hsign	

Fields

Name

Declaration

```
public String Name
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

DTyp

Declaration

```
public WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eDTyp	

RTyp

Declaration

```
public WDataSci.JniPMML.FieldMDEnums.eRTyp RTyp
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eRTyp	

StringMaxLength

Declaration

```
public int StringMaxLength
```

Field Value

TYPE	DESCRIPTION
int	

ByteMaxLength

Declaration

```
public long ByteMaxLength
```

Field Value

TYPE	DESCRIPTION
long	

ByteMemLength

Declaration

```
public long ByteMemLength
```

Field Value

TYPE	DESCRIPTION
long	

ExternalDTyp

Declaration

```
public WDataSci.JniPMML.FieldMDEnums.eDTyp ExternalDTyp
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eDTyp	

ByteMemOffset

Declaration

```
public long ByteMemOffset
```

Field Value

TYPE	DESCRIPTION
long	

Format

Declaration

```
public String Format
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

HDF5DataType

Declaration

```
public WDataSci.JniPMML.WranglerHDF5.HDF5DataType HDF5DataType
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.WranglerHDF5.HDF5DataType	

Methods

Equals(WDataSci.JniPMML.FieldBaseMD arg)

Declaration

```
public boolean Equals(WDataSci.JniPMML.FieldBaseMD arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Returns

TYPE	DESCRIPTION
boolean	

Copy(WDataSci.JniPMML.FieldBaseMD arg)

Declaration

```
public void Copy(WDataSci.JniPMML.FieldBaseMD arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

FLenByteLength()

Declaration

```
public long FLenByteLength()
```

Returns

TYPE	DESCRIPTION
long	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

isVLen()

Declaration

```
public boolean isVLen()
```

Returns

TYPE	DESCRIPTION
boolean	

Consistency()

Declaration

```
public void Consistency()
```

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

TYPE	CONDITION

isMappedToHDF5DataType()

Declaration

```
public boolean isMappedToHDF5DataType()
```

Returns

TYPE	DESCRIPTION
boolean	

MapToHDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp)

Declaration

```
public WDataSci.JniPMML.FieldBaseMD MapToHDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eDTyp	DTyp	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

MapToHDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp, int nStringMaxLength, boolean anyVLen)

Declaration

```
public WDataSci.JniPMML.FieldBaseMD MapToHDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp, int
nStringMaxLength, boolean anyVLen)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eDTyp	DTyp	
int	nStringMaxLength	
boolean	anyVLen	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

MapToHDF5DataType(int hclass, int hlength, int horder, int hsign)

Declaration

```
public WDataSci.JniPMML.FieldBaseMD MapToHDF5DataType(int hclass, int hlength, int horder, int hsign)
```

Parameters

TYPE	NAME	DESCRIPTION
int	hclass	
int	hlength	
int	horder	
int	hsign	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

MapToHDF5DataType(long arg)

Declaration

```
public WDataSci.JniPMML.FieldBaseMD MapToHDF5DataType(long arg)
```

Parameters

TYPE	NAME	DESCRIPTION
long	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

Exceptions

TYPE	CONDITION

Class FieldBaseMD.Default

Inheritance

java.lang.Object

FieldBaseMD.Default

Namespace:

Assembly: .dll

Syntax

```
public static class FieldBaseMD.Default
```

Constructors

Default()

Declaration

```
public Default()
```

Fields

HeaderStringMaxLength

Declaration

```
public static final int HeaderStringMaxLength
```

Field Value

TYPE	DESCRIPTION
int	

StringMaxLength

Declaration

```
public static final int StringMaxLength
```

Field Value

TYPE	DESCRIPTION
int	

Class FieldMD

Inheritance

WDataSci.JniPMML.FieldBaseMD

FieldMD

Implements

com.WDataSci.JniPMML.FieldMDIKey<org.dmg.pmml.FieldName>

Syntax

```
public class FieldMD extends WDataSci.JniPMML.FieldBaseMD implements WDataSci.JniPMML.FieldMDIKey<FieldName>
```

Constructors

FieldMD()

Declaration

```
public FieldMD()
```

FieldMD(WDataSci.JniPMML.FieldMD arg)

Declaration

```
public FieldMD(WDataSci.JniPMML.FieldMD arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

FieldMD(String Name, int hclass, int hlength, int horder, int hsign)

Declaration

```
public FieldMD(String Name, int hclass, int hlength, int horder, int hsign)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	Name	
int	hclass	
int	hlength	
int	horder	
int	hsign	

Fields

MapKey

Declaration

```
public FieldName MapKey
```

Field Value

TYPE	DESCRIPTION
org.dmg.pmml.FieldName	

Methods

MappedKey()

Declaration

```
public FieldName MappedKey()
```

Returns

TYPE	DESCRIPTION
org.dmg.pmml.FieldName	

MappedKeyValue()

Declaration

```
public String MappedKeyValue()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

hasMapKey()

Declaration

```
public boolean hasMapKey()
```

Returns

TYPE	DESCRIPTION
boolean	

MapToMapKey(FieldName aFieldName)

Declaration

```
public void MapToMapKey(FieldName aFieldName)
```

Parameters

TYPE	NAME	DESCRIPTION
org.dmg.pmml.FieldName	aFieldName	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

MapToMapKey(String aFieldStringName)

Declaration

```
public WDataSci.JniPMML.FieldMD MapToMapKey(String aFieldStringName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aFieldStringName	

Returns

TYPE	DESCRIPTION

Equals(WDataSci.JniPMML.FieldMD arg)

Declaration

```
public boolean Equals(WDataSci.JniPMML.FieldMD arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Returns

TYPE	DESCRIPTION
boolean	

Copy(WDataSci.JniPMML.FieldMD arg)

Declaration

```
public void Copy(WDataSci.JniPMML.FieldMD arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

Implements

com.WDataSci.JniPMML.FieldMDIKey<org.dmg.pmml.FieldName>

Class FieldMDEnums

Inheritance

java.lang.Object

FieldMDEnums

Namespace:

Assembly: .dll

Syntax

```
public class FieldMDEnums
```

Constructors

FieldMDEnums()

Declaration

```
public FieldMDEnums()
```

Enum FieldMDEnums.eDTyp

Namespace:

Assembly: .dll

Syntax

```
public enum FieldMDEnums.eDTyp extends Enum<WDataSci.JniPMML.FieldMDEnums.eDTyp>
```

Fields

NAME	DESCRIPTION
Unk	
Dbl	
Lng	
Int	
Dte	
DTm	
Str	
VLS	
Byt	
Bln	

Methods

NAME	DESCRIPTION
values()	
valueOf(String name)	
isString()	
isVLenString()	
isVLen()	
isNumeric()	
AsInt()	
FromInt(int arg)	
FromAlias(String arg, int[] typl)	

NAME	DESCRIPTION
toVerboseString()	
toString()	
ToString()	
bIn(WDataSci.JniPMML.FieldMDEnums.eDTyp[] args)	

Enum FieldMDEnums.eRTyp

Syntax

```
public enum FieldMDEnums.eRTyp extends Enum<WDataSci.JniPMML.FieldMDEnums.eRTyp>
```

Fields

NAME	DESCRIPTION
Unknown	
Output	
Target	
Feature	
Cluster	

Methods

NAME	DESCRIPTION
values()	
valueOf(String name)	
AsInt()	
FromInt(int arg)	
FromAlias(String arg)	
toVerboseString()	
toString()	
ToString()	
bIn(WDataSci.JniPMML.FieldMDEnums.eRTyp[] args)	

Interface FieldMDIKey<T>

Namespace:

Assembly: .dll

Syntax

```
public interface FieldMDIKey<T>
```

Type Parameters

NAME	DESCRIPTION
T	

Methods

MappedKey()

Declaration

```
public abstract T MappedKey()
```

Returns

TYPE	DESCRIPTION
T	

hasMapKey()

Declaration

```
public abstract boolean hasMapKey()
```

Returns

TYPE	DESCRIPTION
boolean	

MapToMapKey(T arg)

Declaration

```
public abstract void MapToMapKey(T arg)
```

Parameters

TYPE	NAME	DESCRIPTION
T	arg	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

Class JniPMML

JniPMML is the main bridge between callers (at the moment, either from C# or a Java command line) and a wrap around a PMML document and its evaluator. To allow a caching of multiple documents and input/output maps, JniPMML contains a collection of JniPMMLItems which can be called by handle or the last one used.

The intent is to pass data frame like blocks of data as inputs for evaluation. Because of incomplete interfaces in HDF5, this is done via a ByteBuffer from C#, but Wranglers for FlatFiles, HDF5, DBB (Direct ByteBuffer) and others are or will be implemented.

Inheritance

java.lang.Object

JniPMML

Namespace:

Assembly: .dll

Syntax

```
public class JniPMML
```

Constructors

JniPMML()

Declaration

```
public JniPMML()
```

Fields

Handle

Declaration

```
protected Integer[] Handle
```

Field Value

TYPE	DESCRIPTION

Items

Declaration

```
protected List<WDataSci.JniPMML.JniPMMLItem> Items
```

Field Value

TYPE	DESCRIPTION
java.util.List<com.WDataSci.JniPMML.JniPMMLItem>	

Methods

isValidHandle(int arg)

Declaration

```
public boolean isValidHandle(int arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg	

Returns

TYPE	DESCRIPTION
boolean	

HandleMajor()

Declaration

```
public int HandleMajor()
```

Returns

TYPE	DESCRIPTION
int	

HandleMajor(int arg)

Declaration

```
public int HandleMajor(int arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

Handle()

Declaration

```
public String Handle()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

Handle(int arg)

Declaration

```
public String Handle(int arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

HandleMinor()

Declaration

```
public int HandleMinor()
```

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

HandleMinor(int arg)

Declaration

```
public int HandleMinor(int arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

HandleNext()

Declaration

```
public int HandleNext()
```

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

ItemNew()

Declaration

```
public WDataSci.JniPMML.JniPMMLItem ItemNew()
```

Returns

TYPE	DESCRIPTION
WDataSci.JniPMML.JniPMMLItem	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

ItemNewHandle()

Declaration

```
public int ItemNewHandle()
```

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

GetItem()

Declaration

```
public WDataSci.JniPMML.JniPMMLItem GetItem()
```

Returns

TYPE	DESCRIPTION
WDataSci.JniPMML.JniPMMLItem	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

GetItem(int arg)

Declaration

```
public WDataSci.JniPMML.JniPMMLItem GetItem(int arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg	

Returns

TYPE	DESCRIPTION
WDataSci.JniPMML.JniPMMLItem	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

ItemDispose(int arg)

Declaration

```
public void ItemDispose(int arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mPMMLLoadFromString(String arg)

Declaration

```
public String mPMMLLoadFromString(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mPMMLLoadFromString(int arg0, String arg)

Declaration

```
public String mPMMLLoadFromString(int arg0, String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mPMMLLoadFromFile(String arg)

Declaration

```
public String mPMMLLoadFromFile(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mPMMLLoadFromFile(int arg0, String arg)

Declaration

```
public String mPMMLLoadFromFile(int arg0, String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

sPMMLLoadedString()

Declaration

```
public String sPMMLLoadedString()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

sPMMLLoadedString(int arg0)

Declaration

```
public String sPMMLLoadedString(int arg0)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

sPMMLLoadedFileName()

Declaration

```
public String sPMMLLoadedFileName()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

sPMMLLoadedFileName(int arg0)

Declaration

```
public String sPMMLLoadedFileName(int arg0)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFromHDF5()

Declaration

```
public int mReadMapFromHDF5()
```

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFromXSDString(String aInputSchemaString)

Declaration

```
public Document mReadMapFromXSDString(String aInputSchemaString)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aInputSchemaString	

Returns

TYPE	DESCRIPTION
org.w3c.dom.Document	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFromXSDString(int arg0, String aInputSchemaString)

Declaration

```
public Document mReadMapFromXSDString(int arg0, String aInputSchemaString)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.lang.String	aInputSchemaString	

Returns

TYPE	DESCRIPTION
org.w3c.dom.Document	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mReadMapFromXSDFile(String aFileName)

Declaration

```
public Document mReadMapFromXSDFile(String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
org.w3c.dom.Document	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mReadMapFromXSDFile(int arg0, String aFileName)

Declaration

```
public Document mReadMapFromXSDFile(int arg0, String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
org.w3c.dom.Document	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mReadMapFromByteBuffer(ByteBuffer arg)

Declaration

```
public int mReadMapFromByteBuffer(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mReadMapFromByteBuffer(int arg0, ByteBuffer arg)

Declaration

```
public String mReadMapFromByteBuffer(int arg0, ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mReadMapFromByteBufferTest(ByteBuffer arg, String aFileName)

Declaration

```
public int mReadMapFromByteBufferTest(ByteBuffer arg, String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mReadMapFromByteBufferTest(int arg0, ByteBuffer arg, String aFileName)

Declaration

```
public int mReadMapFromByteBufferTest(int arg0, ByteBuffer arg, String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.nio.ByteBuffer	arg	
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mMapCheck(String aFileName)

Declaration

```
public String mMapCheck(String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mMapCheck(int arg0, String aFileName)

Declaration

```
public String mMapCheck(int arg0, String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

PMMLEvaluator()

Declaration

```
public Evaluator PMMLEvaluator()
```

Returns

TYPE	DESCRIPTION
org.jpmml.evaluator.Evaluator	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

PMMLEvaluator(int arg0)

Declaration

```
public Evaluator PMMLEvaluator(int arg0)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	

Returns

TYPE	DESCRIPTION
org.jpmml.evaluator.Evaluator	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

PMMLEvaluate(WDataSci.JniPMML.RecordSet aInputRecordSet, boolean bAnySystemOut, boolean bVerboseOutput)

Declaration

```
public List<Map<FieldName,Object>> PMMLEvaluate(WDataSci.JniPMML.RecordSet aInputRecordSet, boolean
bAnySystemOut, boolean bVerboseOutput)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.RecordSet](#)	aInputRecordSet	
boolean	bAnySystemOut	
boolean	bVerboseOutput	

Returns

TYPE	DESCRIPTION
java.util.List<java.util.Map<org.dmg.pmml.FieldName,java.lang.Object>>	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

PMMLEvaluate(int arg0, WDataSci.JniPMML.RecordSet aInputRecordSet, boolean bAnySystemOut, boolean bVerboseOutput)

Declaration

```
public List<Map<FieldName,Object>> PMMLEvaluate(int arg0, WDataSci.JniPMML.RecordSet aInputRecordSet, boolean bAnySystemOut, boolean bVerboseOutput)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
[WDataSci.JniPMML.RecordSet](#)	aInputRecordSet	
boolean	bAnySystemOut	
boolean	bVerboseOutput	

Returns

TYPE	DESCRIPTION
java.util.List<java.util.Map<org.dmg.pmml.FieldName,java.lang.Object>>	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mEvaluateRecordSetAndHoldResults(ByteBuffer arg)

Declaration

```
public int mEvaluateRecordSetAndHoldResults(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mEvaluateRecordSetAndHoldResults(int arg0, ByteBuffer arg)

Declaration

```
public int mEvaluateRecordSetAndHoldResults(int arg0, ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

nRowsOfOutputRecordSet()

Declaration

```
public int nRowsOfOutputRecordSet()
```

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

nRowsOfOutputRecordSet(int arg0)

Declaration

```
public int nRowsOfOutputRecordSet(int arg0)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

nColumnsOfOutputRecordSet()

Declaration

```
public int nColumnsOfOutputRecordSet()
```

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

nColumnsOfOutputRecordSet(int arg0)

Declaration

```
public int nColumnsOfOutputRecordSet(int arg0)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWriteOutputMapToByteBuffer(ByteBuffer arg, int nColumns, int nColumnNameMaxByteLength)

Declaration

```
public String mWriteOutputMapToByteBuffer(ByteBuffer arg, int nColumns, int nColumnNameMaxByteLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
int	nColumns	
int	nColumnNameMaxByteLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mPreRunPrepOutputMap(int nColumnNameMaxByteLength, int nStringMaxLength)

Declaration

```
public int mPreRunPrepOutputMap(int nColumnNameMaxByteLength, int nStringMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	nColumnNameMaxByteLength	
int	nStringMaxLength	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mPreRunPrepOutputMap(int arg0, int nColumnNameMaxByteLength, int nStringMaxLength)

Declaration

```
public int mPreRunPrepOutputMap(int arg0, int nColumnNameMaxByteLength, int nStringMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	

TYPE	NAME	DESCRIPTION
int	nColumnNameMaxByteLength	
int	nStringMaxLength	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mPreRunWriteOutputMapToByteBuffer(ByteBuffer arg, int nColumnNameMaxByteLength, int nStringMaxLength)

Declaration

```
public String mPreRunWriteOutputMapToByteBuffer(ByteBuffer arg, int nColumnNameMaxByteLength, int nStringMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
int	nColumnNameMaxByteLength	
int	nStringMaxLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mPreRunWriteOutputMapToByteBuffer(int arg0, ByteBuffer arg, int nColumnNameMaxByteLength, int nStringMaxLength)

Declaration

```
public String mPreRunWriteOutputMapToByteBuffer(int arg0, ByteBuffer arg, int nColumnNameMaxByteLength, int nStringMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.nio.ByteBuffer	arg	
int	nColumnNameMaxByteLength	
int	nStringMaxLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mWriteOutputMapToByteBuffer(int arg0, ByteBuffer arg, int nColumns, int nColumnNameMaxByteLength)

Declaration

```
public String mWriteOutputMapToByteBuffer(int arg0, ByteBuffer arg, int nColumns, int
nColumnNameMaxByteLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.nio.ByteBuffer	arg	
int	nColumns	
int	nColumnNameMaxByteLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mWriteOutputRecordSetToByteBuffer(ByteBuffer arg, int nRows, long nRecordCoreLength, long nRecordVariableLength, long nCoreLength, long nTotalLength)

Declaration

```
public int mWriteOutputRecordSetToByteBuffer(ByteBuffer arg, int nRows, long nRecordCoreLength, long
nRecordVariableLength, long nCoreLength, long nTotalLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
int	nRows	
long	nRecordCoreLength	
long	nRecordVariableLength	
long	nCoreLength	
long	nTotalLength	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mWriteOutputRecordSetToByteBuffer(int arg0, ByteBuffer arg, int nRows, long nRecordCoreLength, long nRecordVariableLength, long nCoreLength, long nTotalLength)

Declaration

```
public int mWriteOutputRecordSetToByteBuffer(int arg0, ByteBuffer arg, int nRows, long nRecordCoreLength, long
nRecordVariableLength, long nCoreLength, long nTotalLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.nio.ByteBuffer	arg	
int	nRows	
long	nRecordCoreLength	
long	nRecordVariableLength	
long	nCoreLength	

TYPE	NAME	DESCRIPTION
long	nTotalLength	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mEvaluateRecordSetWithFileOutput(ByteBuffer arg, String aFileName, String aFileType, int OutputHDF5FixedStringLength)

Declaration

```
public int mEvaluateRecordSetWithFileOutput(ByteBuffer arg, String aFileName, String aFileType, int
OutputHDF5FixedStringLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
java.lang.String	aFileName	
java.lang.String	aFileType	
int	OutputHDF5FixedStringLength	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mEvaluateRecordSetWithFileOutput(int arg0, ByteBuffer arg, String aFileName, String aFileType, int OutputHDF5FixedStringLength)

Declaration

```
public int mEvaluateRecordSetWithFileOutput(int arg0, ByteBuffer arg, String aFileName, String aFileType, int
OutputHDF5FixedStringLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	arg0	
java.nio.ByteBuffer	arg	
java.lang.String	aFileName	
java.lang.String	aFileType	
int	OutputHDF5FixedStringLength	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mCmdArgsRecap()

Declaration

```
public static String mCmdArgsRecap()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

mCmdRun(String arg)

Declaration

```
public static String mCmdRun(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Class JniPMMLItem

Syntax

```
public class JniPMMLItem
```

Constructors

JniPMMLItem()

Declaration

```
public JniPMMLItem()
```

JniPMMLItem(Integer[] Handle)

Declaration

```
public JniPMMLItem(Integer[] Handle)
```

Parameters

TYPE	NAME	DESCRIPTION
	Handle	

JniPMMLItem(int HandleMajor, int HandleMinor)

Declaration

```
public JniPMMLItem(int HandleMajor, int HandleMinor)
```

Parameters

TYPE	NAME	DESCRIPTION
int	HandleMajor	
int	HandleMinor	

Fields

HandleMajor

Declaration

```
protected Integer HandleMajor
```

Field Value

TYPE	DESCRIPTION
java.lang.Integer	

HandleMinor

Declaration

```
protected Integer HandleMinor
```

Field Value

TYPE	DESCRIPTION
java.lang.Integer	

ConfigMatter

Declaration

```
public WDataSci.JniPMML.JniPMMLItem.__ConfigMatter ConfigMatter
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.JniPMMLItem.__ConfigMatter	

PMMLMatter

Declaration

```
public WDataSci.JniPMML.JniPMMLItem.__PMMLMatter PMMLMatter
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.JniPMMLItem.__PMMLMatter	

InputMatter

Declaration

```
public WDataSci.JniPMML.JniPMMLItem.__InputMatter InputMatter
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.JniPMMLItem.__InputMatter	

OutputMatter

Declaration

```
public WDataSci.JniPMML.JniPMMLItem.__OutputMatter OutputMatter
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.JniPMMLItem.__OutputMatter	

Methods

PreDispose()

Declaration

```
public void PreDispose()
```

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

Reset()

Declaration

```
public void Reset()
```

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

Dispose()

Declaration

```
public void Dispose()
```

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

PMMLLoadFromString(String arg)

Declaration

```
public String PMMLLoadFromString(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

PMMLLoadFromFile(String arg)

Declaration

```
public String PMMLLoadFromFile(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
java.lang.Exception	

PMMLLoadedString()

Declaration

```
public String PMMLLoadedString()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

PMMLLoadedFileName()

Declaration

```
public String PMMLLoadedFileName()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

PMMLDataFields()

Declaration

```
public DataField[] PMMLDataFields()
```

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

PMMLDataFieldNames()

Declaration

```
public FieldName[] PMMLDataFieldNames()
```

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

PMMLDataFieldStringNames()

Declaration

```
public String[] PMMLDataFieldStringNames()
```

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFromHDF5()

Declaration

```
public int mReadMapFromHDF5()
```

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFromXSDString(String aInputSchemaString)

Declaration

```
public Document mReadMapFromXSDString(String aInputSchemaString)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aInputSchemaString	

Returns

TYPE	DESCRIPTION
org.w3c.dom.Document	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFromXSDFile(String aFileName)

Declaration

```
public Document mReadMapFromXSDFile(String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
org.w3c.dom.Document	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFromByteBuffer(ByteBuffer arg)

Declaration

```
public int mReadMapFromByteBuffer(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mReadMapFromByteBufferTest(ByteBuffer arg, String aFileName)

Declaration

```
public int mReadMapFromByteBufferTest(ByteBuffer arg, String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mMapCheck(String aFileName)

Declaration

```
public String mMapCheck(String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION

TYPE	CONDITION
WDataSci.WDS.WDSException	

PMMLEvaluator()

Declaration

```
public Evaluator PMMLEvaluator()
```

Returns

TYPE	DESCRIPTION
org.jpmml.evaluator.Evaluator	

PMMLEvaluate(WDataSci.JniPMML.RecordSet aInputRecordSet, boolean bAnySystemOut, boolean bVerboseOutput)

Declaration

```
public List<Map<FieldName,Object>> PMMLEvaluate(WDataSci.JniPMML.RecordSet aInputRecordSet, boolean
bAnySystemOut, boolean bVerboseOutput)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSet	aInputRecordSet	
boolean	bAnySystemOut	
boolean	bVerboseOutput	

Returns

TYPE	DESCRIPTION
java.util.List<java.util.Map<org.dmg.pmml.FieldName,java.lang.Object>>	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mEvaluateRecordSetAndHoldResults(ByteBuffer arg)

Declaration

```
public int mEvaluateRecordSetAndHoldResults(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

nRowsOfOutputRecordSet()

Declaration

```
public int nRowsOfOutputRecordSet()
```

Returns

TYPE	DESCRIPTION
int	

nColumnsOfOutputRecordSet()

Declaration

```
public int nColumnsOfOutputRecordSet()
```

Returns

TYPE	DESCRIPTION
int	

mPreRunPrepOutputMap(int nColumnNameMaxByteLength, int nStringMaxLength)

Declaration

```
public int mPreRunPrepOutputMap(int nColumnNameMaxByteLength, int nStringMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
int	nColumnNameMaxByteLength	
int	nStringMaxLength	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mPreRunWriteOutputMapToByteBuffer(ByteBuffer arg, int nColumnNameMaxByteLength, int nStringMaxLength)

Declaration

```
public String mPreRunWriteOutputMapToByteBuffer(ByteBuffer arg, int nColumnNameMaxByteLength, int
nStringMaxLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
int	nColumnNameMaxByteLength	
int	nStringMaxLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWriteOutputMapToByteBuffer(ByteBuffer arg, int nColumns, int nColumnNameMaxByteLength)

Declaration

```
public String mWriteOutputMapToByteBuffer(ByteBuffer arg, int nColumns, int nColumnNameMaxByteLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
int	nColumns	
int	nColumnNameMaxByteLength	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWriteOutputRecordSetToByteBuffer(ByteBuffer arg, int nRows, long nRecordCoreLength, long nRecordVariableLength, long nCoreLength, long nTotalLength)

```
public int mWriteOutputRecordSetToByteBuffer(ByteBuffer arg, int nRows, long nRecordCoreLength, long
nRecordVariableLength, long nCoreLength, long nTotalLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
int	nRows	
long	nRecordCoreLength	
long	nRecordVariableLength	
long	nCoreLength	
long	nTotalLength	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mEvaluateRecordSetWithFileOutput(ByteBuffer arg, String aFileName, String aFileType, int OutputHDF5FixedStringLength)

Declaration

```
public int mEvaluateRecordSetWithFileOutput(ByteBuffer arg, String aFileName, String aFileType, int
OutputHDF5FixedStringLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
java.lang.String	aFileName	
java.lang.String	aFileType	
int	OutputHDF5FixedStringLength	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

Class JniPMMLItem.__ConfigMatter

Inheritance

java.lang.Object

JniPMMLItem.__ConfigMatter

Namespace:

Assembly: .dll

Syntax

```
public class JniPMMLItem.__ConfigMatter
```

Constructors

__ConfigMatter()

Declaration

```
public __ConfigMatter()
```

Fields

InternalString

Declaration

```
public String InternalString
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

Class JniPMMLItem.__InputMatter

Inheritance

java.lang.Object

JniPMMLItem.__InputMatter

Namespace:

Assembly: .dll

Syntax

```
public class JniPMMLItem.__InputMatter
```

Constructors

__InputMatter()

Declaration

```
public __InputMatter()
```

Fields

RecordSetMD

Declaration

```
public WDataSci.JniPMML.RecordSetMD RecordSetMD
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	

RecordSet

Declaration

```
public WDataSci.JniPMML.RecordSet RecordSet
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.RecordSet	

_XSDDoc

Declaration

```
public Document _XSDDoc
```

Field Value

TYPE	DESCRIPTION
org.w3c.dom.Document	

_XSDFileName

Declaration

```
public String _XSDFileName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

_XSDString

Declaration

```
public String _XSDString
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

Class JniPMMLItem.__OutputMatter

Inheritance

java.lang.Object

JniPMMLItem.__OutputMatter

Namespace:

Assembly: .dll

Syntax

```
public class JniPMMLItem.__OutputMatter
```

Constructors

__OutputMatter()

Declaration

```
public __OutputMatter()
```

Fields

RecordSetMD

Declaration

```
public WDataSci.JniPMML.RecordSetMD RecordSetMD
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	

RecordSet

Declaration

```
public WDataSci.JniPMML.RecordSet RecordSet
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.RecordSet	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

Class JniPMMLItem.__PMMLMatter

Inheritance

java.lang.Object

JniPMMLItem.__PMMLMatter

Namespace:

Assembly: .dll

Syntax

```
public class JniPMMLItem.__PMMLMatter
```

Constructors

__PMMLMatter()

Declaration

```
public __PMMLMatter()
```

Fields

Doc

Declaration

```
protected PMML Doc
```

Field Value

TYPE	DESCRIPTION
org.dmg.pmml.PMML	

Evaluator

Declaration

```
protected ModelEvaluator Evaluator
```

Field Value

TYPE	DESCRIPTION
org.jpmml.evaluator.ModelEvaluator	

_XMLString

Declaration

```
public String _XMLString
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

_XMLFileName

Declaration

```
public String _XMLFileName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

Methods

Dispose()

Declaration

```
public void Dispose()
```

```
public String _XMLFileName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

Class RecordSet

Inheritance

java.lang.Object

RecordSet

Namespace:

Assembly: .dll

Syntax

```
public class RecordSet
```

Constructors

RecordSet()

Declaration

```
public RecordSet()
```

RecordSet(List<Map<FieldName,Object>> _Records)

Declaration

```
public RecordSet(List<Map<FieldName,Object>> _Records)
```

Parameters

TYPE	NAME	DESCRIPTION
java.util.List<java.util.Map<org.dmg.pmml.FieldName,java.lang.Object>>	_Records	

Fields

Records

Declaration

```
public List<Map<FieldName,Object>> Records
```

Field Value

TYPE	DESCRIPTION
java.util.List<java.util.Map<org.dmg.pmml.FieldName,java.lang.Object>>	

Records_Orig

Declaration

```
public List<Object[]> Records_Orig
```

Field Value

TYPE	DESCRIPTION
java.util.List<java.lang.Object[]>	

isInput

Declaration

```
public Boolean isInput
```

Field Value

TYPE	DESCRIPTION
java.lang.Boolean	

Methods

Dispose()

Declaration

```
public void Dispose()
```

cAsInput()

Declaration

```
public WDataSci.JniPMML.RecordSet cAsInput()
```

Returns

TYPE	DESCRIPTION

cAsOutput()

Declaration

```
public WDataSci.JniPMML.RecordSet cAsOutput()
```

Returns

TYPE	DESCRIPTION

cAsOutput(List<Map<FieldName,Object>> _Records)

Declaration

```
public WDataSci.JniPMML.RecordSet cAsOutput(List<Map<FieldName,Object>> _Records)
```

Parameters

TYPE	NAME	DESCRIPTION
java.util.List<java.util.Map<org.dmg.pmml.FieldName,java.lang.Object>>	_Records	

Returns

TYPE	DESCRIPTION

isEmpty()

Declaration

```
public Boolean isEmpty()
```

Returns

TYPE	DESCRIPTION
java.lang.Boolean	

mReadRecordSet(WDataSci.JniPMML.RecordSetMD aRecordSetMD)

Declaration

```
public WDataSci.JniPMML.RecordSet mReadRecordSet(WDataSci.JniPMML.RecordSetMD aRecordSetMD)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mWriteRecordSet(WDataSci.JniPMML.RecordSetMD aOutputRecordSetMD, WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet)

Declaration

```
public WDataSci.JniPMML.RecordSet mWriteRecordSet(WDataSci.JniPMML.RecordSetMD aOutputRecordSetMD,
WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aOutputRecordSetMD	
WDataSci.JniPMML.RecordSetMD	aInputRecordSetMD	
	aInputRecordSet	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

TYPE	CONDITION
java.lang.Exception	

Class RecordSetMD

Inheritance

java.lang.Object

RecordSetMD

Namespace:

Assembly: .dll

Syntax

```
public class RecordSetMD
```

Constructors

RecordSetMD(WDataSci.JniPMML.RecordSetMDEnums.eMode arg)

Declaration

```
public RecordSetMD(WDataSci.JniPMML.RecordSetMDEnums.eMode arg)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMDEnums.eMode	arg	

Fields

Mode

Declaration

```
public WDataSci.JniPMML.RecordSetMDEnums.eMode Mode
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.RecordSetMDEnums.eMode	

Type

Declaration

```
public WDataSci.JniPMML.RecordSetMDEnums.eType Type
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.RecordSetMDEnums.eType	

SchemaType

Declaration

```
public WDataSci.JniPMML.RecordSetMDEnums.eSchemaType SchemaType
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.RecordSetMDEnums.eSchemaType	

Column

Declaration

```
public WDataSci.JniPMML.FieldMD[] Column
```

Field Value

TYPE	DESCRIPTION

FileMatter

Declaration

```
public WDataSci.JniPMML.WranglerFlatFile FileMatter
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.WranglerFlatFile	

SchemaMatter

Declaration

```
public WDataSci.JniPMML.RecordSetMD.__SchemaMatter SchemaMatter
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.RecordSetMD.__SchemaMatter	

ModeMatter

Declaration

```
public WDataSci.JniPMML.RecordSetMD.__ModeMatter ModeMatter
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.RecordSetMD.__ModeMatter	

HDF5Matter

Declaration

```
public WDataSci.JniPMML.WranglerHDF5 HDF5Matter
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.WranglerHDF5	

DBBMatter

Declaration

```
public WDataSci.JniPMML.WranglerDBB DBBMatter
```

Field Value

TYPE	DESCRIPTION
WDataSci.JniPMML.WranglerDBB	

DefaultHeaderMaxStringLength

Declaration

```
public static long DefaultHeaderMaxStringLength
```

Field Value

TYPE	DESCRIPTION
long	

DefaultHeaderMaxStringByteLength

Declaration

```
public static long DefaultHeaderMaxStringByteLength
```

Field Value

TYPE	DESCRIPTION
long	

DefaultMaxStringLength

Declaration

```
public static long DefaultMaxStringLength
```

Field Value

TYPE	DESCRIPTION
long	

DefaultMaxStringByteLength

Declaration

```
public static long DefaultMaxStringByteLength
```

Field Value

TYPE	DESCRIPTION
long	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cUsingCmdArguments(WDataSci.JniPMML.CmdArgs args)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cUsingCmdArguments(WDataSci.JniPMML.CmdArgs args)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.CmdArgs](#)	args	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

nColumns()

Declaration

```
public int nColumns()
```

Returns

TYPE	DESCRIPTION
int	

nColumns(WDataSci.JniPMML.FieldMDEnums.eRTyp arg)

Declaration

```
public int nColumns(WDataSci.JniPMML.FieldMDEnums.eRTyp arg)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eRTyp	arg	

Returns

TYPE	DESCRIPTION
int	

cFromFile(String aFileName)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cFromFile(String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION

cToFile(String aFileName)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cToFile(String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION

cAs(WDataSci.JniPMML.RecordSetMDEnums.eType arg, WDataSci.JniPMML.RecordSetMDEnums.eSchemaType schema)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cAs(WDataSci.JniPMML.RecordSetMDEnums.eType arg,
WDataSci.JniPMML.RecordSetMDEnums.eSchemaType schema)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.RecordSetMDEnums.eType](#)	arg	
[WDataSci.JniPMML.RecordSetMDEnums.eSchemaType](#)	schema	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cAs(WDataSci.JniPMML.RecordSetMDEnums.eType arg, WDataSci.JniPMML.RecordSetMDEnums.eSchemaType schema, Boolean isFileName, String schemadetails)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cAs(WDataSci.JniPMML.RecordSetMDEnums.eType arg,
WDataSci.JniPMML.RecordSetMDEnums.eSchemaType schema, Boolean isFileName, String schemadetails)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.RecordSetMDEnums.eType](#)	arg	
[WDataSci.JniPMML.RecordSetMDEnums.eSchemaType](#)	schema	
java.lang.Boolean	isFileName	
java.lang.String	schemadetails	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cAs(WDataSci.JniPMML.RecordSetMDEnums.eType arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cAs(WDataSci.JniPMML.RecordSetMDEnums.eType arg)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMDEnums.eType	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cWithHeaderRow()

Declaration

```
public WDataSci.JniPMML.RecordSetMD cWithHeaderRow()
```

Returns

TYPE	DESCRIPTION

cWithCompositeFieldNameDlm(String aCompositeFieldNameDlm)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cWithCompositeFieldNameDlm(String aCompositeFieldNameDlm)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aCompositeFieldNameDlm	

Returns

TYPE	DESCRIPTION

cAsDlmFile(String aFileName)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cAsDlmFile(String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION

cAsDlmFile(String aFileName, String dlm)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cAsDlmFile(String aFileName, String dlm)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aFileName	
java.lang.String	dlm	

Returns

TYPE	DESCRIPTION

cWithOutRepeatInputSet()

Declaration

```
public WDataSci.JniPMML.RecordSetMD cWithOutRepeatInputSet()
```

Returns

TYPE	DESCRIPTION

cRepeatInputSet()

Declaration

```
public WDataSci.JniPMML.RecordSetMD cRepeatInputSet()
```

Returns

TYPE	DESCRIPTION

cRepeatInputSetWithSuffix(String aInputFieldSuffix)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cRepeatInputSetWithSuffix(String aInputFieldSuffix)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aInputFieldSuffix	

Returns

TYPE	DESCRIPTION

cRepeatInputSetWithSuffix(String aInputFieldSuffix, String aCompositeFieldNameDlm)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cRepeatInputSetWithSuffix(String aInputFieldSuffix, String
aCompositeFieldNameDlm)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aInputFieldSuffix	
java.lang.String	aCompositeFieldNameDlm	

Returns

TYPE	DESCRIPTION

cWithDlm(String arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cWithDlm(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

cWithDelimiter(String arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cWithDelimiter(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

nHeaderStringMaxLength()

Declaration

```
public long nHeaderStringMaxLength()
```

Returns

TYPE	DESCRIPTION
long	

nHeaderByteMaxLength()

Declaration

```
public long nHeaderByteMaxLength()
```

Returns

TYPE	DESCRIPTION
long	

isModeValid()

Declaration

```
public boolean isModeValid()
```

Returns

TYPE	DESCRIPTION
boolean	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cSetHeaderBufferAs(WDataSci.JniPMML.DBB arg, int nRecords, int nRecordCoreLength, int nRecordVariableLength)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetHeaderBufferAs(WDataSci.JniPMML.DBB arg, int nRecords, int
nRecordCoreLength, int nRecordVariableLength)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.DBB	arg	
int	nRecords	
int	nRecordCoreLength	
int	nRecordVariableLength	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cSetHeaderBufferFrom(WDataSci.JniPMML.DBB arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetHeaderBufferFrom(WDataSci.JniPMML.DBB arg)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.DBB	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cSetHeaderBufferAs(ByteBuffer arg, int nRecords, int nRecordCoreLength, int nRecordVariableLength)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetHeaderBufferAs(ByteBuffer arg, int nRecords, int nRecordCoreLength,
int nRecordVariableLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
int	nRecords	
int	nRecordCoreLength	
int	nRecordVariableLength	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cSetHeaderBufferFrom(ByteBuffer arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetHeaderBufferFrom(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cSetRecordSetBufferAs(ByteBuffer arg, long nRecords, long nRecordCoreLength, long nRecordVariableLength, long nCoreLength, long nTotalLength)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetRecordSetBufferAs(ByteBuffer arg, long nRecords, long
nRecordCoreLength, long nRecordVariableLength, long nCoreLength, long nTotalLength)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	
long	nRecords	
long	nRecordCoreLength	
long	nRecordVariableLength	
long	nCoreLength	
long	nTotalLength	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cSetRecordSetBufferAs(ByteBuffer arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetRecordSetBufferAs(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cSetRecordSetBufferFrom(ByteBuffer arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetRecordSetBufferFrom(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cSetRecordSetBufferAs(WDataSci.JniPMML.DBB arg, long nRecords, long nRecordCoreLength, long nRecordVariableLength, long nCoreLength, long nTotalLength)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetRecordSetBufferAs(WDataSci.JniPMML.DBB arg, long nRecords, long nRecordCoreLength, long nRecordVariableLength, long nCoreLength, long nTotalLength)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.DBB](#)	arg	
long	nRecords	
long	nRecordCoreLength	
long	nRecordVariableLength	
long	nCoreLength	
long	nTotalLength	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cSetRecordSetBufferAs(WDataSci.JniPMML.DBB arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetRecordSetBufferAs(WDataSci.JniPMML.DBB arg)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.DBB	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cSetRecordSetBufferFrom(WDataSci.JniPMML.DBB arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cSetRecordSetBufferFrom(WDataSci.JniPMML.DBB arg)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.DBB	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cWithRecordSetElementName(String arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cWithRecordSetElementName(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION

cWithRecordSetAndRecordElementNames(String arg, String arg1)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cWithRecordSetAndRecordElementNames(String arg, String arg1)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	
java.lang.String	arg1	

Returns

TYPE	DESCRIPTION

cWithRecordElementName(String arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cWithRecordElementName(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION

cWithDataSetName(String arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD cWithDataSetName(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mCopyColumnsFrom(WDataSci.JniPMML.RecordSetMD arg)

Declaration

```
public WDataSci.JniPMML.RecordSetMD mCopyColumnsFrom(WDataSci.JniPMML.RecordSetMD arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

Equals(WDataSci.JniPMML.RecordSetMD arg, boolean bIgnoreMode)

Declaration

```
public boolean Equals(WDataSci.JniPMML.RecordSetMD arg, boolean bIgnoreMode)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	
boolean	bIgnoreMode	

Returns

TYPE	DESCRIPTION
boolean	

mReadMapFor(WDataSci.JniPMML.JniPMMLItem aJniPMMLItem, PrintWriter pw, boolean bFillDictionaryNames)

Declaration

```
public WDataSci.JniPMML.RecordSetMD mReadMapFor(WDataSci.JniPMML.JniPMMLItem aJniPMMLItem, PrintWriter pw,
boolean bFillDictionaryNames)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.JniPMMLItem](#)	aJniPMMLItem	
java.io.PrintWriter	pw	
boolean	bFillDictionaryNames	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mColumnConsistency()

Declaration

```
public WDataSci.JniPMML.RecordSetMD mColumnConsistency()
```

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mPrepForOutput(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMML, List<Map<FieldName,Object>> Results)

Declaration

```
public WDataSci.JniPMML.RecordSetMD mPrepForOutput(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD,
WDataSci.JniPMML.JniPMMLItem aJniPMML, List<Map<FieldName,Object>> Results)
```

Parameters

TYPE	NAME	DESCRIPTION
	aInputRecordSetMD	
[WDataSci.JniPMML.JniPMMLItem](#)	aJniPMML	
java.util.List<java.util.Map<org.dmg.pmml.FieldName,java.lang.Object>>	Results	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
java.lang.Exception	

mUpdateWithPMMLSchema(WDataSci.JniPMML.JniPMMLItem aJniPMML)

Declaration

```
public void mUpdateWithPMMLSchema(WDataSci.JniPMML.JniPMMLItem aJniPMML)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.JniPMMLItem	aJniPMML	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWriteMapToBuffer()

Declaration

```
public int mWriteMapToBuffer()
```

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWriteMapToBuffer(ByteBuffer arg)

Declaration

```
public int mWriteMapToBuffer(ByteBuffer arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.nio.ByteBuffer	arg	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

WDataSci.WDS.WDSException

Class RecordSetMD.__ModeMatter

Inheritance

java.lang.Object

RecordSetMD.__ModeMatter

Syntax

```
public class RecordSetMD.__ModeMatter
```

Constructors

__ModeMatter()

Declaration

```
public __ModeMatter()
```

Fields

nInputFields

Declaration

```
public int nInputFields
```

Field Value

TYPE	DESCRIPTION
int	

bRepeatInputFields

Declaration

```
public boolean bRepeatInputFields
```

Field Value

TYPE	DESCRIPTION
boolean	

CompositeNameDlm

Declaration

```
public String CompositeNameDlm
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

CompositeInputNameSuffix

Declaration

```
public String CompositeInputNameSuffix
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

OutputMaxStringLength

Declaration

```
public int OutputMaxStringLength
```

Field Value

TYPE	DESCRIPTION
int	

Methods

Equals(WDataSci.JniPMML.RecordSetMD.__ModeMatter arg)

Declaration

```
public boolean Equals(WDataSci.JniPMML.RecordSetMD.__ModeMatter arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Returns

TYPE	DESCRIPTION
boolean	

Class RecordSetMD.__SchemaMatter

Inheritance

WDataSci.JniPMML.WranglerXSD

RecordSetMD.__SchemaMatter

Namespace:

Assembly: .dll

Syntax

```
public class RecordSetMD.__SchemaMatter extends WDataSci.JniPMML.WranglerXSD
```

Constructors

__SchemaMatter()

Declaration

```
public __SchemaMatter()
```

Fields

InputSchemaFileName

Declaration

```
public String InputSchemaFileName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

InputSchemaString

Declaration

```
public String InputSchemaString
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

InputSchema

Declaration

```
public Document InputSchema
```

Field Value

TYPE	DESCRIPTION
org.w3c.dom.Document	

RecordSetElementName

Declaration

```
public String RecordSetElementName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

RecordElementName

Declaration

```
public String RecordElementName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

Methods

Equals(WDataSci.JniPMML.RecordSetMD.__SchemaMatter arg)

Declaration

```
public boolean Equals(WDataSci.JniPMML.RecordSetMD.__SchemaMatter arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Returns

TYPE	DESCRIPTION
boolean	

Dispose()

Declaration

```
public void Dispose()
```

Class RecordSetMDEnums

Inheritance

java.lang.Object

RecordSetMDEnums

Namespace:

Assembly: .dll

Syntax

```
public class RecordSetMDEnums
```

Constructors

RecordSetMDEnums()

Declaration

```
public RecordSetMDEnums()
```

Enum RecordSetMDEnums.eMode

Namespace:

Assembly: .dll

Syntax

```
public enum RecordSetMDEnums.eMode extends Enum<WDataSci.JniPMML.RecordSetMDEnums.eMode>
```

Fields

NAME	DESCRIPTION
Unknown	
Input	
Output	
Internal	

Methods

NAME	DESCRIPTION
values()	
valueOf(String name)	
Equals(WDataSci.JniPMML.RecordSetMDEnums.eMode arg)	
bIn(WDataSci.JniPMML.RecordSetMDEnums.eMode[] args)	
toString()	

Enum RecordSetMDEnums.eSchemaType

Namespace:

Assembly: .dll

Syntax

```
public enum RecordSetMDEnums.eSchemaType extends Enum<WDataSci.JniPMML.RecordSetMDEnums.eSchemaType>
```

Fields

NAME	DESCRIPTION
Unknown	
XSD	
XML	
JSON	
HDF5	
DBB	
RecordSetMD	
SQL	
NamingConvention	

Methods

NAME	DESCRIPTION
values()	
valueOf(String name)	
bIn(WDataSci.JniPMML.RecordSetMDEnums.eSchemaType[] args)	
FromInt(int arg)	
FromAlias(String arg)	
toString()	

Enum RecordSetMDEnums.eType

Syntax

```
public enum RecordSetMDEnums.eType extends Enum<WDataSci.JniPMML.RecordSetMDEnums.eType>
```

Fields

NAME	DESCRIPTION
Unknown	
Dlm	
CSV	
TXT	
HDF5	
DBB	
XML	
JSON	
SQL	

Methods

NAME	DESCRIPTION
values()	
valueOf(String name)	
isFlatFile()	
isFile()	
FromInt(int arg)	
FromAlias(String arg)	
toString()	
bIn(WDataSci.JniPMML.RecordSetMDEnums.eType[] args)	
bIn(String[] arg)	

Class Util

Inheritance

java.lang.Object

Util

Namespace:

Assembly: .dll

Syntax

```
public class Util
```

Constructors

Util()

Declaration

```
public Util()
```

Methods

RecordSingleName(String arg)

Declaration

```
public static String RecordSingleName(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

RecordSetElementName(Document xInputSchema)

RecordSetElementName - Returns the name attribute of the RecordSet element of the InputSchema

The RecordSet element of the input schema contains zero or more single rows of data fields.

Unless specified elsewhere, the convention used here is a singular term for a single row (such as Record, Row, Vector, or Observation), and a plural form for the container of multiple rows (such as RecordSet, Records, Rows or RowSet).

The schema of the data fields provides the cross map between the input data set and the PMML data dictionary. The input fields can queried from the InputSchema under the assumption that the XPath query follows the concept:

> /RecordSet/Record/*Fields*

The RecordSetElementName function extracts the corresponding name from the InputSchema.

When not provided as an input, the convention used here is to have a schema which is consistent with what can be easily used for the exportable XMLMap of a ListObject in Excel.

A simple form of such an XML has the pattern:

> > > > *value1* > > > > *value2* > > > > *value3* > >

However, the XMLSchema associated with a simple structure looks like:

<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xi="http://www.w3.org/2001/XInclude" attributeFormDefault="unqualified" elementFormDefault="qualified">

<xs:element name="RecordSet">

<xs:complexType> <xs:sequence>

<xs:element name="Record" maxOccurs="unbounded">

<xs:complexType> <xs:sequence>

<xs:element name="Field1" type="?"/>

<xs:element name="Field2" type="?"/>

<xs:element name="Field3" type="?"/>

</xs:sequence> </xs:complexType>

</xs:element>

</xs:sequence> </xs:complexType>

</xs:element>

</xs:schema>

Declaration

```
public static String RecordSetElementName(Document xInputSchema)
```

Parameters

TYPE	NAME	DESCRIPTION
org.w3c.dom.Document	xInputSchema	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

Class WranglerDBB

Inheritance

java.lang.Object

WranglerDBB

Syntax

```
public class WranglerDBB
```

Constructors

WranglerDBB()

Declaration

```
public WranglerDBB()
```

Fields

Header

Declaration

```
public WDataSci.JniPMML.WranglerDBB.__DBBMatter Header
```

Field Value

TYPE	DESCRIPTION
[WDataSci.JniPMML.WranglerDBB.__DBBMatter](#)	

RecordSet

Declaration

```
public WDataSci.JniPMML.WranglerDBB.__DBBMatter RecordSet
```

Field Value

TYPE	DESCRIPTION
[WDataSci.JniPMML.WranglerDBB.__DBBMatter](#)	

Methods

Dispose()

Declaration

```
public void Dispose()
```

isValid()

Declaration

```
public boolean isValid()
```

Returns

TYPE	DESCRIPTION
boolean	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cSetHeaderBufferAs(WDataSci.JniPMML.DBB arg, int nRecords, int nRecordCoreLength, int nRecordVariableLength)

Declaration

```
public WDataSci.JniPMML.WranglerDBB cSetHeaderBufferAs(WDataSci.JniPMML.DBB arg, int nRecords, int
nRecordCoreLength, int nRecordVariableLength)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.DBB	arg	
int	nRecords	
int	nRecordCoreLength	
int	nRecordVariableLength	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

cSetHeaderBufferFrom(WDataSci.JniPMML.DBB arg)

Declaration

```
public WDataSci.JniPMML.WranglerDBB cSetHeaderBufferFrom(WDataSci.JniPMML.DBB arg)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.DBB	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cSetRecordSetBufferAs(WDataSci.JniPMML.DBB arg)

Declaration

```
public WDataSci.JniPMML.WranglerDBB cSetRecordSetBufferAs(WDataSci.JniPMML.DBB arg)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.DBB](#)	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cSetRecordSetBufferAs(WDataSci.JniPMML.DBB arg, long nRecords, long nRecordCoreLength, long nRecordVariableLength, long nCoreLength, long nTotalLength)

Declaration

```
public WDataSci.JniPMML.WranglerDBB cSetRecordSetBufferAs(WDataSci.JniPMML.DBB arg, long nRecords, long nRecordCoreLength, long nRecordVariableLength, long nCoreLength, long nTotalLength)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.DBB](#)	arg	
long	nRecords	
long	nRecordCoreLength	
long	nRecordVariableLength	
long	nCoreLength	

TYPE	NAME	DESCRIPTION
long	nTotalLength	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

cSetRecordSetBufferFrom(WDataSci.JniPMML.DBB arg)

Declaration

```
public WDataSci.JniPMML.WranglerDBB cSetRecordSetBufferFrom(WDataSci.JniPMML.DBB arg)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.DBB](#)	arg	

Returns

TYPE	DESCRIPTION

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

mReadMap(WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMML, PrintWriter pw, boolean bFillDictionaryNames)

Declaration

```
public void mReadMap(WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMML,
PrintWriter pw, boolean bFillDictionaryNames)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.RecordSetMD](#)	aRecordSetMD	
[WDataSci.JniPMML.JniPMMLItem](#)	aJniPMML	

TYPE	NAME	DESCRIPTION
java.io.PrintWriter	pw	
boolean	bFillDictionaryNames	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadRecordSet(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet, PrintWriter pw)

Declaration

```
public void mReadRecordSet(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet
aInputRecordSet, PrintWriter pw)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aInputRecordSetMD	
WDataSci.JniPMML.RecordSet	aInputRecordSet	
java.io.PrintWriter	pw	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWriteMap(WDataSci.JniPMML.RecordSetMD aRecordSetMD)

Declaration

```
public int mWriteMap(WDataSci.JniPMML.RecordSetMD aRecordSetMD)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mBytesRequired(WDataSci.JniPMML.RecordSetMD aRecordSetMD, long nRecords, long[] csize, long[] hsize, long[] rsize, long[] cleadsize, long[] hleadsize, long[] hflensize, long[] hvlensize, long[] rleadsize, long[] rflensize, long[] rvlensize)

Declaration

```
public void mBytesRequired(WDataSci.JniPMML.RecordSetMD aRecordSetMD, long nRecords, long[] csize, long[]
hsize, long[] rsize, long[] cleadsize, long[] hleadsize, long[] hflensize, long[] hvlensize, long[] rleadsize,
long[] rflensize, long[] rvlensize)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	
long	nRecords	
	csize	
	hsize	
	rsize	
	cleadsize	
	hleadsize	
	hflensize	
	hvlensize	
	rleadsize	
	rflensize	
	rvlensize	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWritePrepFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, long nRecords)

Declaration

```
public void mWritePrepFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, long nRecords)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.RecordSetMD](#)	aRecordSetMD	
long	nRecords	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

mWriteRecordSet(WDataSci.JniPMML.RecordSetMD outRecordSetMD, WDataSci.JniPMML.RecordSet aOutputRecordSet, WDataSci.JniPMML.RecordSetMD inRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet)

Declaration

```
public int mWriteRecordSet(WDataSci.JniPMML.RecordSetMD outRecordSetMD, WDataSci.JniPMML.RecordSet
aOutputRecordSet, WDataSci.JniPMML.RecordSetMD inRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet)
```

Parameters

TYPE	NAME	DESCRIPTION
[WDataSci.JniPMML.RecordSetMD](#)	outRecordSetMD	
[WDataSci.JniPMML.RecordSet](#)	aOutputRecordSet	
[WDataSci.JniPMML.RecordSetMD](#)	inRecordSetMD	
[WDataSci.JniPMML.RecordSet](#)	aInputRecordSet	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	

Class WranglerDBB.__DBBMatter

Inheritance

java.lang.Object

WranglerDBB.__DBBMatter

Namespace:

Assembly: .dll

Syntax

```
public class WranglerDBB.__DBBMatter
```

Constructors

__DBBMatter()

Declaration

```
public __DBBMatter()
```

Fields

Buffer

Declaration

```
public WDataSci.JniPMML.DBB Buffer
```

Field Value

TYPE	DESCRIPTION
[WDataSci.JniPMML.DBB](#)	

MaxStringLength

Declaration

```
public long MaxStringLength
```

Field Value

TYPE	DESCRIPTION
long	

MaxStringByteLength

Declaration

```
public long MaxStringByteLength
```

Field Value

TYPE	DESCRIPTION
long	

bIsManagedInJava

Declaration

```
public Boolean bIsManagedInJava
```

Field Value

TYPE	DESCRIPTION
java.lang.Boolean	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Class WranglerFlatFile

Inheritance

java.lang.Object

WranglerFlatFile

Namespace:

Assembly: .dll

Syntax

```
public class WranglerFlatFile
```

Constructors

WranglerFlatFile()

Declaration

```
public WranglerFlatFile()
```

Fields

FileName

Declaration

```
public String FileName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

Path

Declaration

```
public String Path
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

Dlm

Declaration

```
public String Dlm
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

hasHeaderRow

Declaration

```
public boolean hasHeaderRow
```

Field Value

TYPE	DESCRIPTION
boolean	

__CSVParser

Declaration

```
public CSVParser __CSVParser
```

Field Value

TYPE	DESCRIPTION
org.apache.commons.csv.CSVParser	

__CSVParserIterator

Declaration

```
public Iterator<CSVRecord> __CSVParserIterator
```

Field Value

TYPE	DESCRIPTION
java.util.Iterator<org.apache.commons.csv.CSVRecord>	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Equals(WDataSci.JniPMML.WranglerFlatFile arg)

Declaration

```
public boolean Equals(WDataSci.JniPMML.WranglerFlatFile arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Returns

TYPE	DESCRIPTION
boolean	

cPointToFile(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, String aFileName, boolean hasHeaderRow, String dlm)

Declaration

```
public void cPointToFile(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, String aFileName, boolean
hasHeaderRow, String dlm)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aInputRecordSetMD	
java.lang.String	aFileName	
boolean	hasHeaderRow	
java.lang.String	dlm	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMML, PrintWriter pw, boolean bFillDictionaryNames)

Declaration

```
public void mReadMapFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMML,
PrintWriter pw, boolean bFillDictionaryNames)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	
WDataSci.JniPMML.JniPMMLItem	aJniPMML	
java.io.PrintWriter	pw	
boolean	bFillDictionaryNames	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadRecordSet(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet, PrintWriter pw)

Declaration

```
public void mReadRecordSet(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet
aInputRecordSet, PrintWriter pw)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aInputRecordSetMD	
WDataSci.JniPMML.RecordSet	aInputRecordSet	
java.io.PrintWriter	pw	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadRecordSet(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet, String aPath, String aFileName, boolean hasHeaderRow, String dlm, PrintWriter pw)

Declaration

```
public void mReadRecordSet(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet
aInputRecordSet, String aPath, String aFileName, boolean hasHeaderRow, String dlm, PrintWriter pw)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aInputRecordSetMD	
WDataSci.JniPMML.RecordSet	aInputRecordSet	
java.lang.String	aPath	
java.lang.String	aFileName	
boolean	hasHeaderRow	
java.lang.String	dlm	
java.io.PrintWriter	pw	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWriteRecordSet(WDataSci.JniPMML.RecordSetMD aOutputRecordSetMD, WDataSci.JniPMML.RecordSet aOutputRecordSet, WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet)

Declaration

```
public void mWriteRecordSet(WDataSci.JniPMML.RecordSetMD aOutputRecordSetMD, WDataSci.JniPMML.RecordSet
aOutputRecordSet, WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aOutputRecordSetMD	
WDataSci.JniPMML.RecordSet	aOutputRecordSet	
WDataSci.JniPMML.RecordSetMD	aInputRecordSetMD	
WDataSci.JniPMML.RecordSet	aInputRecordSet	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWriteRecordSet(WDataSci.JniPMML.RecordSetMD aOutputRecordSetMD, WDataSci.JniPMML.RecordSet aOutputRecordSet)

Declaration

```
public void mWriteRecordSet(WDataSci.JniPMML.RecordSetMD aOutputRecordSetMD, WDataSci.JniPMML.RecordSet
aOutputRecordSet)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aOutputRecordSetMD	
WDataSci.JniPMML.RecordSet	aOutputRecordSet	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

Class WranglerHDF5

Syntax

```
public class WranglerHDF5
```

Constructors

WranglerHDF5()

Declaration

```
public WranglerHDF5()
```

Fields

File

Declaration

```
public H5File File
```

Field Value

TYPE	DESCRIPTION
hdf.object.h5.H5File	

CompoundDS

Declaration

```
public H5CompoundDS CompoundDS
```

Field Value

TYPE	DESCRIPTION
hdf.object.h5.H5CompoundDS	

DSName

Declaration

```
public String DSName
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

bIsInMemory

Declaration

```
public boolean bIsInMemory
```

Field Value

TYPE	DESCRIPTION
boolean	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

finalize()

Declaration

```
protected void finalize()
```

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMML, PrintWriter pw, boolean bFillDictionaryNames)

Declaration

```
public void mReadMapFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMML,
PrintWriter pw, boolean bFillDictionaryNames)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	
WDataSci.JniPMML.JniPMMLItem	aJniPMML	
java.io.PrintWriter	pw	
boolean	bFillDictionaryNames	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadPrepFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, PrintWriter pw)

Declaration

```
public long mReadPrepFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, PrintWriter pw)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	
java.io.PrintWriter	pw	

Returns

TYPE	DESCRIPTION
long	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mWritePrepFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, PrintWriter pw)

Declaration

```
public long mWritePrepFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, PrintWriter pw)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	
java.io.PrintWriter	pw	

Returns

TYPE	DESCRIPTION
long	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

UpdateOutputMapForHDF5(WDataSci.JniPMML.RecordSetMD aRecordSetMD, PrintWriter pw)

Declaration

```
public void UpdateOutputMapForHDF5(WDataSci.JniPMML.RecordSetMD aRecordSetMD, PrintWriter pw)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	
java.io.PrintWriter	pw	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mReadRecordSet(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet, PrintWriter pw)

Declaration

```
public void mReadRecordSet(WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet
aInputRecordSet, PrintWriter pw)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aInputRecordSetMD	
WDataSci.JniPMML.RecordSet	aInputRecordSet	
java.io.PrintWriter	pw	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

mWriteMap(WDataSci.JniPMML.RecordSetMD aRecordSetMD)

Declaration

```
public int mWriteMap(WDataSci.JniPMML.RecordSetMD aRecordSetMD)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	

Returns

TYPE	DESCRIPTION
int	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mWriteRecordSet(WDataSci.JniPMML.RecordSetMD aOutputRecordSetMD, WDataSci.JniPMML.RecordSet aOutputRecordSet, WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet)

Declaration

```
public void mWriteRecordSet(WDataSci.JniPMML.RecordSetMD aOutputRecordSetMD, WDataSci.JniPMML.RecordSet
aOutputRecordSet, WDataSci.JniPMML.RecordSetMD aInputRecordSetMD, WDataSci.JniPMML.RecordSet aInputRecordSet)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aOutputRecordSetMD	
WDataSci.JniPMML.RecordSet	aOutputRecordSet	
WDataSci.JniPMML.RecordSetMD	aInputRecordSetMD	
WDataSci.JniPMML.RecordSet	aInputRecordSet	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

new_HDF5DataType(int hclass, int hlength, int horder, int hsign)

Declaration

```
public WDataSci.JniPMML.WranglerHDF5.HDF5DataType new_HDF5DataType(int hclass, int hlength, int horder, int
hsign)
```

Parameters

TYPE	NAME	DESCRIPTION
int	hclass	
int	hlength	
int	horder	
int	hsign	

Returns

TYPE	DESCRIPTION
WDataSci.JniPMML.WranglerHDF5.HDF5DataType	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

new_HDF5DataType(WDataSci.JniPMML.WranglerHDF5.HDF5DataType arg)

Declaration

```
public WDataSci.JniPMML.WranglerHDF5.HDF5DataType new_HDF5DataType(WDataSci.JniPMML.WranglerHDF5.HDF5DataType arg)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.WranglerHDF5.HDF5DataType	arg	

Returns

TYPE	DESCRIPTION
WDataSci.JniPMML.WranglerHDF5.HDF5DataType	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

new_HDF5DataType(WDataSci.JniPMML.FieldBaseMD arg)

Declaration

```
public WDataSci.JniPMML.WranglerHDF5.HDF5DataType new_HDF5DataType(WDataSci.JniPMML.FieldBaseMD arg)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.FieldBaseMD	arg	

Returns

TYPE	DESCRIPTION
WDataSci.JniPMML.WranglerHDF5.HDF5DataType	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

new_HDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp)

Declaration

```
public WDataSci.JniPMML.WranglerHDF5.HDF5DataType new_HDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eDTyp	DTyp	

Returns

TYPE	DESCRIPTION
WDataSci.JniPMML.WranglerHDF5.HDF5DataType	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	
java.lang.Exception	

new_HDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp, long nStringMaxLength, boolean anyVLen)

Declaration

```
public WDataSci.JniPMML.WranglerHDF5.HDF5DataType new_HDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp,
long nStringMaxLength, boolean anyVLen)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eDTyp	DTyp	
long	nStringMaxLength	
boolean	anyVLen	

Returns

TYPE	DESCRIPTION
WDataSci.JniPMML.WranglerHDF5.HDF5DataType	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

new_HDF5DataType(long arg)

Declaration

```
public WDataSci.JniPMML.WranglerHDF5.HDF5DataType new_HDF5DataType(long arg)
```

Parameters

TYPE	NAME	DESCRIPTION
long	arg	

Returns

TYPE	DESCRIPTION
[WDataSci.JniPMML.WranglerHDF5.HDF5DataType](#)	

Exceptions

TYPE	CONDITION
[WDataSci.WDS.WDSException](#)	
java.lang.Exception	

Class WranglerHDF5.HDF5DataType

Inheritance

java.lang.Object

WranglerHDF5.HDF5DataType

Syntax

```
public class WranglerHDF5.HDF5DataType
```

Constructors

HDF5DataType(int hclass, int hlength, int horder, int hsign)

Declaration

```
public HDF5DataType(int hclass, int hlength, int horder, int hsign)
```

Parameters

TYPE	NAME	DESCRIPTION
int	hclass	
int	hlength	
int	horder	
int	hsign	

HDF5DataType(Datatype arg)

Declaration

```
public HDF5DataType(Datatype arg)
```

Parameters

TYPE	NAME	DESCRIPTION
hdf.object.Datatype	arg	

HDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp, long nStringMaxLength, boolean anyVLen)

Declaration

```
public HDF5DataType(WDataSci.JniPMML.FieldMDEnums.eDTyp DTyp, long nStringMaxLength, boolean anyVLen)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eDTyp	DTyp	
long	nStringMaxLength	
boolean	anyVLen	

Fields

data

Declaration

```
public Datatype data
```

Field Value

TYPE	DESCRIPTION
hdf.object.Datatype	

Methods

eDTyp()

Declaration

```
public WDataSci.JniPMML.FieldMDEnums.eDTyp eDTyp()
```

Returns

TYPE	DESCRIPTION
WDataSci.JniPMML.FieldMDEnums.eDTyp	

Equals(WDataSci.JniPMML.WranglerHDF5.HDF5DataType arg)

Declaration

```
public boolean Equals(WDataSci.JniPMML.WranglerHDF5.HDF5DataType arg)
```

Parameters

TYPE	NAME	DESCRIPTION
	arg	

Returns

TYPE	DESCRIPTION
boolean	

Class WranglerXSD

Inheritance

java.lang.Object

WranglerXSD

Namespace:

Assembly: .dll

Syntax

```
public class WranglerXSD
```

Constructors

WranglerXSD()

Declaration

```
public WranglerXSD()
```

Methods

mReadMapFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMMLItem, PrintWriter pw, boolean bFillDictionaryNames)

Declaration

```
public void mReadMapFor(WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMMLItem,
PrintWriter pw, boolean bFillDictionaryNames)
```

Parameters

TYPE	NAME	DESCRIPTION
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	
WDataSci.JniPMML.JniPMMLItem	aJniPMMLItem	
java.io.PrintWriter	pw	
boolean	bFillDictionaryNames	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

mReadMapFor(Document aDoc, WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem aJniPMMLItem, PrintWriter pw, Boolean bFillDictionaryNames)

Declaration

```
public void mReadMapFor(Document aDoc, WDataSci.JniPMML.RecordSetMD aRecordSetMD, WDataSci.JniPMML.JniPMMLItem
aJniPMMLItem, PrintWriter pw, Boolean bFillDictionaryNames)
```

Parameters

TYPE	NAME	DESCRIPTION
org.w3c.dom.Document	aDoc	
WDataSci.JniPMML.RecordSetMD	aRecordSetMD	
WDataSci.JniPMML.JniPMMLItem	aJniPMMLItem	
java.io.PrintWriter	pw	
java.lang.Boolean	bFillDictionaryNames	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

XSDHeader()

Declaration

```
public static String XSDHeader()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

XSDRecordSet_Open(String rns, String rn)

Declaration

```
public static String XSDRecordSet_Open(String rns, String rn)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	rns	
java.lang.String	rn	

Returns

TYPE	DESCRIPTION
java.lang.String	

XSDRecordSet_Close()

Declaration

```
public static String XSDRecordSet_Close()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

XSDFooter()

Declaration

```
public static String XSDFooter()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

XSDColumn(String name, String dtyp)

Declaration

```
public static String XSDColumn(String name, String dtyp)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	name	
java.lang.String	dtyp	

Returns

TYPE	DESCRIPTION
java.lang.String	

XSDTypes()

Declaration

```
public static String XSDTypes()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

WDS-Java

General utilitities that independent of JniPMML code. To simplify assemblies and jars, this code is included in the larger projects, but is also compiled as a stand alone.

On the java side, there is a separate WDS-00.00.00.jar generated but it is pulled into a shaded jar so that only one WDS-JniPMML-00.00.00.jar needs to be used in practice.

Namespace com.WDataSci.WDS

Compiled versions of this package may include open source projects under other licenses and detail will included as required in attached resources, although the source code here is generally to be released under the license above either publicly or to clients of Wypasek Data Science, Inc.

Classes

Util

WDSException

Class Util

Inheritance

java.lang.Object

Util

Namespace:

Assembly: .dll

Syntax

```
public class Util
```

Constructors

Util()

Declaration

```
public Util()
```

Methods

FetchFileAsString(String arg)

Declaration

```
public static String FetchFileAsString(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
WDataSci.WDS.WDSException	

bIn(String arg0, String[] args)

Declaration

```
public static boolean bIn(String arg0, String[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg0	
	args	

Returns

TYPE	DESCRIPTION
boolean	

MatchingNullity(String A, String B)

Declaration

```
public static boolean MatchingNullity(String A, String B)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	A	
java.lang.String	B	

Returns

TYPE	DESCRIPTION
boolean	

MatchingNullityAndValueEquals(String A, String B)

Declaration

```
public static boolean MatchingNullityAndValueEquals(String A, String B)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	A	
java.lang.String	B	

Returns

TYPE	DESCRIPTION
boolean	

MatchingNullity(Object A, Object B)

Declaration

```
public static boolean MatchingNullity(Object A, Object B)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.Object	A	
java.lang.Object	B	

Returns

TYPE	DESCRIPTION
boolean	

CleanAsNMToken(String arg)

CleanAsNMToken returns a clean and valid NMToken (name token) string for a given input

Following XMLSchema data types, a NMToken cannot contain single or double quotes, or commas. These characters are stripped from the input and any leading, trailing, or interior spaces are removed. The primary characters are Java Regular Expression *word* characters (A-Z_a-z0-9), period, underscore, colon, and dash. It technically also includes CombiningChars and Extenders, but the regular expression here does not implement those in this version.

Declaration

```
public static String CleanAsNMToken(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
java.lang.Exception	

CleanAsToken(String arg)

CleanAsToken returns a clean and valid XMLSchema string data type Token for a given input.

Following XMLSchema data types, a Token does not have leading or trailing spaces, tabs, carriage returns, linefeeds, and interior multiple space sequences are converted to single spaces. CleanAsToken maps all non-printable characters to space before conversion.

Declaration

```
public static String CleanAsToken(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

TYPE	CONDITION
java.lang.Exception	

CleanQuotes(String arg)

CleanQuotes returns a string with double or single quotes removed.

Declaration

```
public static String CleanQuotes(String arg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
java.lang.Exception	

CleanAsString(String arg, String regex_exp, String regex_repl)

CleanAsString (overloaded) allows the regular expression and the replaceAll target to be inputs (for testing)

Declaration

```
public static String CleanAsString(String arg, String regex_exp, String regex_repl)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	arg	
java.lang.String	regex_exp	
java.lang.String	regex_repl	

Returns

TYPE	DESCRIPTION
java.lang.String	

Exceptions

TYPE	CONDITION
java.lang.Exception	

BaseDirAndPath(String aBaseDir, String aPath)

Declaration

```
public static String BaseDirAndPath(String aBaseDir, String aPath)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aBaseDir	
java.lang.String	aPath	

Returns

TYPE	DESCRIPTION
java.lang.String	

PathAndName(String aPath, String aFileName)

Declaration

```
public static String PathAndName(String aPath, String aFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	aPath	
java.lang.String	aFileName	

Returns

TYPE	DESCRIPTION
java.lang.String	

Class WDSException

Inheritance

java.lang.Exception

WDSException

Namespace:

Assembly: .dll

Syntax

```
public class WDSException extends Exception
```

Constructors

WDSException(String msg)

Declaration

```
public WDSException(String msg)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	msg	

WDSException(String msg, Throwable e)

Declaration

```
public WDSException(String msg, Throwable e)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	msg	
java.lang.Throwable	e	

WDSException(String msg, Exception e)

Declaration

```
public WDSException(String msg, Exception e)
```

Parameters

TYPE	NAME	DESCRIPTION
java.lang.String	msg	
java.lang.Exception	e	

Fields

__Message

Declaration

```
public String __Message
```

Field Value

TYPE	DESCRIPTION
java.lang.String	

Methods

getMessage()

Declaration

```
public String getMessage()
```

Returns

TYPE	DESCRIPTION
java.lang.String	

toString()

Declaration

```
public String toString()
```

Returns

TYPE	DESCRIPTION
java.lang.String	