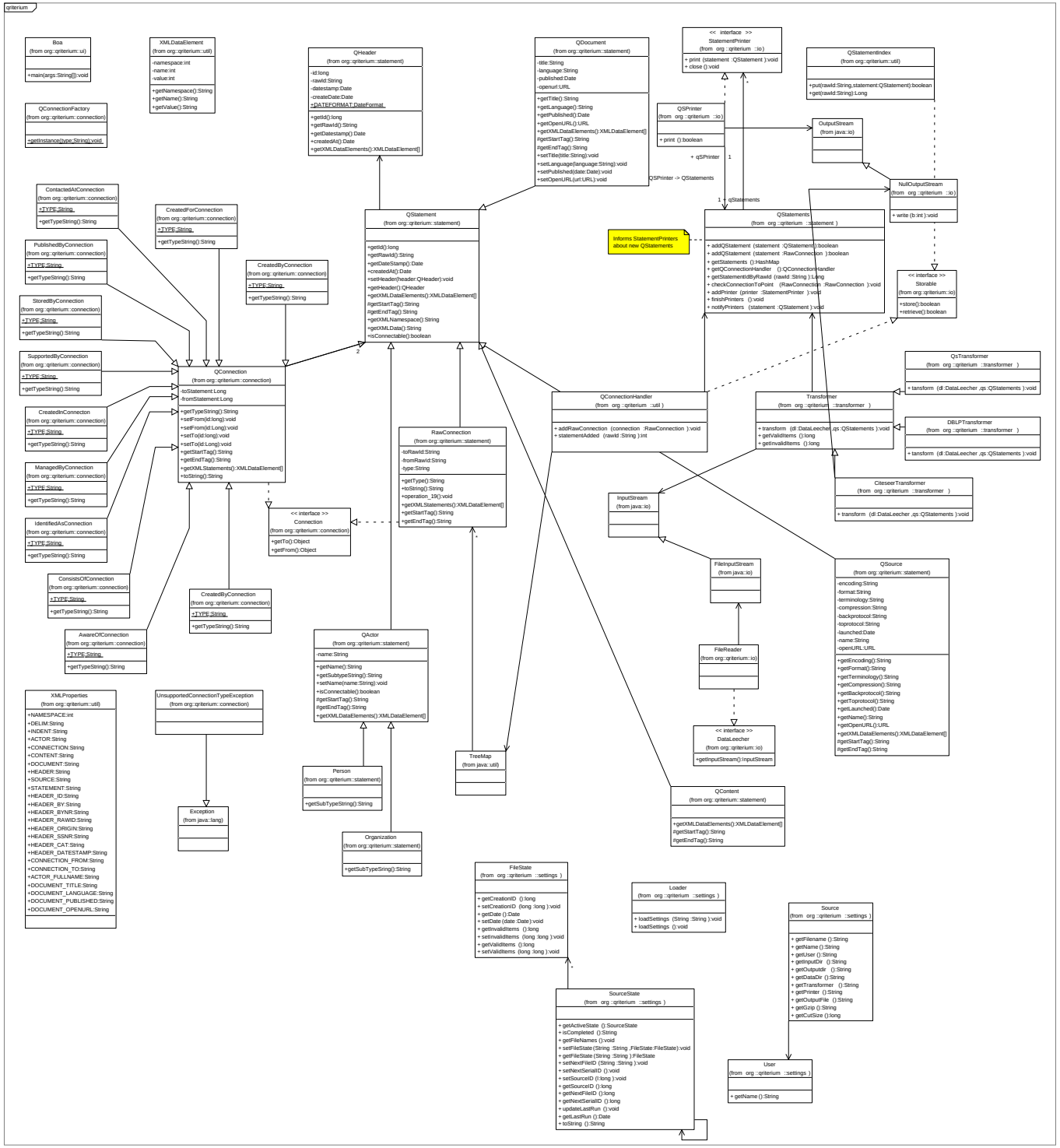
qriterium
Boa
(from org::qriterium::ui)
+main(args:String[]):void
XMLDataElement
(from org::qriterium::util)
-namespace:int
-name:int
-value:int
+getNamespace():String
+getName():String
+getValue():String
QHeader
(from org::qriterium::statement)
-id:long
-rawId:String
-datestamp:Date
-createDate:Date
+DATEFORMAT:DateFormat
+getId():long
+getRawId():String
+getDatestamp():Date
+createdAt():Date
+getXMLDataElements():XMLDataElement[]
QDocument
(from org::qriterium::statement)
-title:String
-language:String
-published:Date
-openurl:URL
+getTitle():String
+getLanguage():String
+getPublished():Date
+getOpenURL():URL
+getXMLDataElements():XMLDataElement[]
#getStartTag():String
#getEndTag():String
+setTitle(title:String):void
+setLanguage(language:String):void
+setPublished(date:Date):void
+setOpenURL(url:URL):void
<< interface >>
StatementPrinter
(from org::qriterium::io)
+print (statement :QStatement):void
+close ():void
QStatementIndex
(from org::qriterium::util)
+put(rawId:String,statement:QStatement):boolean
+get(rawId:String):Long
QSPrinter
(from org::qriterium::io)
+print ():boolean
OutputStream
(from java::io)
+ qSPrinter
1
QSPrinter -> QStatements
NullOutputStream
(from org::qriterium::io)
+write (b:int):void
QConnectionFactory
(from org::qriterium::connection)
+getInstance(type:String):void
ContactedAtConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
CreatedForConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
PublishedByConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
CreatedByConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
StoredByConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
QStatement
(from org::qriterium::statement)
+getId():long
+getRawId():String
+getDateStamp():Date
+createdAt():Date
+setHeader(header:QHeader):void
+getHeader():QHeader
+getXMLDataElements():XMLDataElement[]
#getStartTag():String
#getEndTag():String
+getXMLNamespace():String
+getXMLData():String
+isConnectable():boolean
1 + qStatements
QStatements
(from org::qriterium::statement)
+addQStatement (statement :QStatement):boolean
+addQStatement (statement :RawConnection):boolean
+getStatements ():HashMap
+getQConnectionHandler ():QConnectionHandler
+getStatementIdByRawId (rawId :String):Long
+checkConnectionToPoint (RawConnection :RawConnection):void
+addPrinter (printer :StatementPrinter):void
+finishPrinters ():void
+notifyPrinters (statement :QStatement):void
Informs StatementPrinters about new QStatements
<< interface >>
Storable
(from org::qriterium::io)
+store():boolean
+retrieve():boolean
SupportedByConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
QConnection
(from org::qriterium::connection)
-toStatement:Long
-fromStatement:Long
+getTypeString():String
+setFrom(id:long):void
+setFrom(id:Long):void
+setTo(id:long):void
+setTo(id:Long):void
+getStartTag():String
+getEndTag():String
+getXMLStatements():XMLDataElement[]
+toString():String
2
QsTransformer
(from org::qriterium::transformer)
+tansform (dl:DataLeecher ,qs:QStatements):void
QConnectionHandler
(from org::qriterium::util)
+addRawConnection (connection :RawConnection):void
+statementAdded (rawId :String):int
Transformer
(from org::qriterium::transformer)
+transform (dl:DataLeecher ,qs:QStatements):void
+getValidItems ():long
+getInvalidItems ():long
DBLPTransformer
(from org::qriterium::transformer)
+tansform (dl:DataLeecher ,qs:QStatements):void
CreatedInConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
RawConnection
(from org::qriterium::statement)
-toRawId:String
-fromRawId:String
-type:String
+getType():String
+toString():String
+operation_19():void
+getXMLStatements():XMLDataElement[]
+getStartTag():String
+getEndTag():String
CiteseerTransformer
(from org::qriterium::transformer)
+transform (dl:DataLeecher ,qs:QStatements):void
ManagedByConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
<< interface >>
Connection
(from org::qriterium::connection)
+getTo():Object
+getFrom():Object
InputStream
(from java::io)
IdentifiedAsConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
FileInputStream
(from java::io)
QSource
(from org::qriterium::statement)
-encoding:String
-format:String
-terminology:String
-compression:String
-backprotocol:String
-toprotocol:String
-launched:Date
-name:String
-openURL:URL
+getEncoding():String
+getFormat():String
+getTerminology():String
+getCompression():String
+getBackprotocol():String
+getToprotocol():String
+getLaunched():Date
+getName():String
+getOpenURL():URL
+getXMLDataElements():XMLDataElement[]
#getStartTag():String
#getEndTag():String
ConsistsOfConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
CreatedByConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
AwareOfConnection
(from org::qriterium::connection)
+TYPE:String
+getTypeString():String
QActor
(from org::qriterium::statement)
-name:String
+getName():String
+getSubtypeString():String
+setName(name:String):void
+isConnectable():boolean
#getStartTag():String
#getEndTag():String
+getXMLDataElements():XMLDataElement[]
FileReader
(from org::qriterium::io)
<< interface >>
DataLeecher
(from org::qriterium::io)
+getInputStream():InputStream
XMLProperties
(from org::qriterium::util)
+NAMESPACE:int
+DELIM:String
+INDENT:String
+ACTOR:String
+CONNECTION:String
+CONTENT:String
+DOCUMENT:String
+HEADER:String
+SOURCE:String
+STATEMENT:String
+HEADER_ID:String
+HEADER_BY:String
+HEADER_BYNR:String
+HEADER_RAWID:String
+HEADER_ORIGIN:String
+HEADER_SSNR:String
+HEADER_CAT:String
+HEADER_DATESTAMP:String
+CONNECTION_FROM:String
+CONNECTION_TO:String
+ACTOR_FULLNAME:String
+DOCUMENT_TITLE:String
+DOCUMENT_LANGUAGE:String
+DOCUMENT_PUBLISHED:String
+DOCUMENT_OPENURL:String
UnsupportedConnectionTypeException
(from org::qriterium::connection)
Exception
(from java::lang)
Person
(from org::qriterium::statement)
+getSubTypeString():String
TreeMap
(from java::util)
Organization
(from org::qriterium::statement)
+getSubTypeString():String
QContent
(from org::qriterium::statement)
+getXMLDataElements():XMLDataElement[]
#getStartTag():String
#getEndTag():String
FileState
(from org::qriterium::settings)
+getCreationID ():long
+setCreationID (long :long):void
+getDate ():Date
+setDate (date:Date):void
+getInvalidItems ():long
+setInvalidItems (long :long):void
+getValidItems ():long
+setValidItems (long :long):void
Loader
(from org::qriterium::settings)
+loadSettings (String :String):void
+loadSettings ():void
Source
(from org::qriterium::settings)
+getFilename ():String
+getName ():String
+getUser ():String
+getInputDir ():String
+getOutputdir ():String
+getDataDir ():String
+getTransformer ():String
+getPrinter ():String
+getOutputFile ():String
+getGzip ():String
+getCutSize ():long
SourceState
(from org::qriterium::settings)
+getActiveState ():SourceState
+isCompleted ():String
+getFileNames ():void
+setFileState (String :String ,FileState:FileState):void
+getFileState (String :String):FileState
+setNextFileID (String :String):void
+setNextSerialID ():void
+setSourceID (l:long):void
+getSourceID ():long
+getNextFileID ():long
+getNextSerialID ():long
+updateLastRun ():void
+getLastRun ():Date
+toString ():String
User
(from org::qriterium::settings)
+getName ():String