

```

UMNFFeature
id: String
name: Hashtable
titles: Vector
values: Vector
geometry: GeomElementDouble
<<create>> UMNFeature(id: String)
getEnvelope(): Double
getTitles(): Vector
getValues(): Vector

UMNLayer
visible: boolean
name: String
<<create>> UMNLayer(name: String, isVisible: boolean)
<<create>> UMNLayer(name: String, isVisible: boolean, isQueryable: boolean)
isActive(): boolean
selectable: boolean
setField( Field ): void
setField(VectorField): Vector
void
getFields(): Vector
setUMNFeatureVector(umfFeatures: Vector): void
addUMNFeature(feature: UMNFeature): void
getUMNFeatures(): Vector
<<create>> UMNLayer(name: String)
setName(name: String): void
getName(): String
setQueryable(queryable: boolean): void
setVisible(visible: boolean): void
isVisible(): boolean
isQueryable(): boolean
createXMLRepresentation(): StringBuffer
toXML(): String

```

```

UMNRasterDataSourceHandler
tagContent: StringBuffer
hasException: boolean
exception: Exception
ds: UMNRasterDataSource
layerHandler: UMNLayerHandler
serverURL: String
umfServiceName: String
imageType: String
layerList: Vector
<<create>> UMNRasterDataSourceHandler()
characters(ch: char[], start: int, len: int): void
startElement(namespace: String, localname: String, qname: String, attrs: Attributes): void
endElement(namespace: String, localname: String, qname: String): void
createDataSource(): void
getException(): Exception
getDataSource(): DataSource

```

```

UMNXMLEFeatureContentHandler
source: UMNFeatureResponse
currentLayer: UMNLayer
layerName: String
fieldName: String
id: int
<<create>> UMNXMLFeatureContentHandler(source: UMNFeatureResponse)
characters(ch: char[], start: int, len: int): void
startElement(namespace: String, localname: String, qname: String, attrs: Attributes): void
endElement(namespace: String, localname: String, qname: String): void

```

```

UMNRasterDataSource
IMAGE_JPG: String
IMAGE_PNG: String
IMAGE_GIF: String
mapServerURL: String
configFileURL: String
isInitialized: boolean
umfConfigResp: UMNConfigResponse
templateFile: String
imageType: String
layerBounds: Double
uniqueIDs: Vector
featureFiles: Vector
fields: Vector
respFeature: UMNFeatureResponse
serverURL: Vector
source: String
worldFrame: Double
tempoGraf: File
gfi: GeorasterSystem
<<create>> UMNRasterDataSource()
init(): DataSource
getSources(): String
createUMNRasterDataSource(mapServerURL: String, configFileURL: String): UMNRasterDataSource
<<create>> UMNRasterDataSource(mapServerURL: String, configFileURL: String)
getConfigResponse(): UMNConfigResponse
setLayerList(layerList: Vector): void
addQueryableLayer(layer: UMNLayer): void
getQueryableLayers(): Vector
setGeorasterSystem(gfi: GeorasterSystem): void
setImageType(imageType: String): void
setLayerList(layerList: Vector): void
getWorldFrame(): Double
isExisting(): boolean
getClipWordRec: Double, screenRec: Rectangle2D): BufferedImage
getEnvelope(): Envelope
delete(): boolean
getRecord(id: String): Record
getFields(): Vector
getLayers(): Vector
getIntersectingElementDs(rec: Double): Vector
removeLayer(layer: UMNLayer): void
addLayer(layer: UMNLayer): void
close(): void
createUMNRequest(): void
createUMNFeatureRequest(rec: Double): UMNRequest
sendFeatureRequest(umfRequest: UMNRequest): UMNFeatureResponse
isInitialized(): boolean
getConfigrationFileConfigFileURLPath: String): UMNConfigResponse
sendRequest(umfRequest: UMNRequest): void
getLayerBounds(): Double
setWorldFrame(rec: Double): void
createScaledImage(scaleFactor: double): void
createScaleOptions: RenderedImage, factor: float): RenderedOp
createXMLRepresentation(): StringBuffer
toXML(): String
getScaledImage(scaleFactor: double, clip: Rectangle2D): RenderedImage

```

```

UMNXMLEContentHandler
source: UMNConfigResponse
currentLayer: UMNLayer
layerName: String
isVisible: boolean
useIdentify: boolean
header: String
template: String
footer: String
isQueryable: boolean
<<create>> UMNXMLContentHandler(source: UMNConfigResponse)
characters(ch: char[], start: int, len: int): void
startElement(namespace: String, localname: String, qname: String, attrs: Attributes): void
endElement(namespace: String, localname: String, qname: String): void

```

