

Link attributes

`ex:itemTag = "item1,item2,item3...."`

(REQUIRED)

Specifies what tags in an xml document specifies tags

`ex:setPropertyAsLabel = "propertyOfItem1, propertyOfItem2..."`

(REQUIRED)

Specifies which property should be label of item

`item1.label = propertyOfItem1`

DEFAULT: 1st property of item

`ex:parentRelation = "parentRelation1,parentRelation2..."`

`childItem.parentRelation = parentItemLabel`

`ex:itemType = "typeOfItem1,typeOfItem2...."`

Sets the item type of item1,item2 and so on...

`ex:propertyTags = "propertyTag1,propertyTag2...."`

Used in conjunction with `ex:propertyNames`

Renames `propertyTag1` as `propertyName1`

`ex:propertyNames = "propertyName1,propertyName2...."`

Used in conjunction with `ex:propertyTags`

Notes

This XML importer only deals with xml data that uses nodes as the primary data medium. Tag attributes are not included.

User specifies important items with the `ex:itemTag`. All other nodes are either unimportant (and ignored) or properties. See Rules.

Each attribute is a comma-separated list of values. The first elements of each list are mutual.

The `ex:itemTag` and `ex:setPropertyAsLabel` are required for this importer to work.

Rules

The following set of rules pertains to when an item is found. The program test the child nodes of that item in the following ways:

If child node has no children (only text), then that child node is a property of the enclosing item.

If child node is an item as designated by `ex:itemTag`, then create parent relation between items

If child node has children, but is not an item designated by `ex:itemTag`, then ignore. Instead check the children of the child node with the same rules as specified above.