**Process 0 - Name and title processing – r1**

**0.1) Name and name/title field components**

Name part: all subfields before $t except $e, $4, $h

Role part: $e, $4

Title part: all subfields after $t except $h,, $v,x,y,z,w,0-3,5-8

Series part: $v (8XX) , $x (7XX, 8XX)

Subject components: $x (6XX), $y, $z, $v (6XX)

Genre part: $h

Relationship part: $e/$4 (6XX)

Other?

**0.2) Figuring out Relationship**

If 1XX, then name/title is the resource being described

If 6XX, then relationship is bf:subject

If also $e or $4, then carry over content using bf:relationship property

If 7XX, then

If I2=2, relationship is bf:hasPart

Else relationship is bf:relatedTo

If also have $i, then carry over $i content using bf:relationship property

If 8XX, then relationship bf:series

**0.3 Some Basic RDF Patterns**

**RDF for names**

<resource> bf:contribution [ a bf:Contribution ;

bf:agent [ a bf: Person, Organization, etc.

rdfs:label “label from Process 1.3”;

identifiedBy [

rdf:value “content after (…) in $0 or $w”;

bf:assigner “content of (...) in $0 or $w” ] ;

bflc:nameXXMatchKey “normalized string from Process 1.1”;

bflc:nameXXMarcKey “normalized string from Process 1.2” ] ;

bf:role [rdfs:label “…” ] see Process 1.4

bf:code [rdf:value “…” ] ; see Process 1.4

bf:source id relators URI ] ].

Create MADSRDF stub description

bflc:affiliation “content of $u”

**RDF for titles**

Construct Title class from title subfield; keep Title subproperties in same order as in field.

bf:Work bf:title [

a bf:WorkTitle

rdfs:label “label from Process 2.3” ;

bf:mainTitle “content of $a (X30, 240) or $t (X00, X10, X11)” ;

bf:partnumber “content of $n” ;

bf:partName “content of $p” ;

bflc:titleXXMatchKey “normalized content from Process 2.1” ;

bflc:titleXXMarcKey “normalized content from Process 2.2” ;

bflc:titleSort “sort content from Process 2.4” ] .

identifiedBy [

rdf:value “content after (…) in $0 or $w” ; bf:asssigner “content of (…) before number in $0 or $w” ].

Convert content of other MARC title subfields listed in rdfs:label as specified in spec; order not necessary to preserve.

**RDF for relationships**

<resource> bf:relatedTo URI . or [bnode

URI or bnode a bf:Work ;

rdfs:label “label from Process 2.3”;

identifiedBy [

rdf:value “content after (…) in $0 or $w”;

bf:assigner [rdfs:label “content of (...) in $0 or $w” ] ] ;

bflc:titleXXMatchKey “normalized string from Process 1.1”;

bflc:titleXXMarcKey “normalized string from Process 1.2”.

bf:relationship [rdfs:label from Process 0.5;

bf:source URI ].