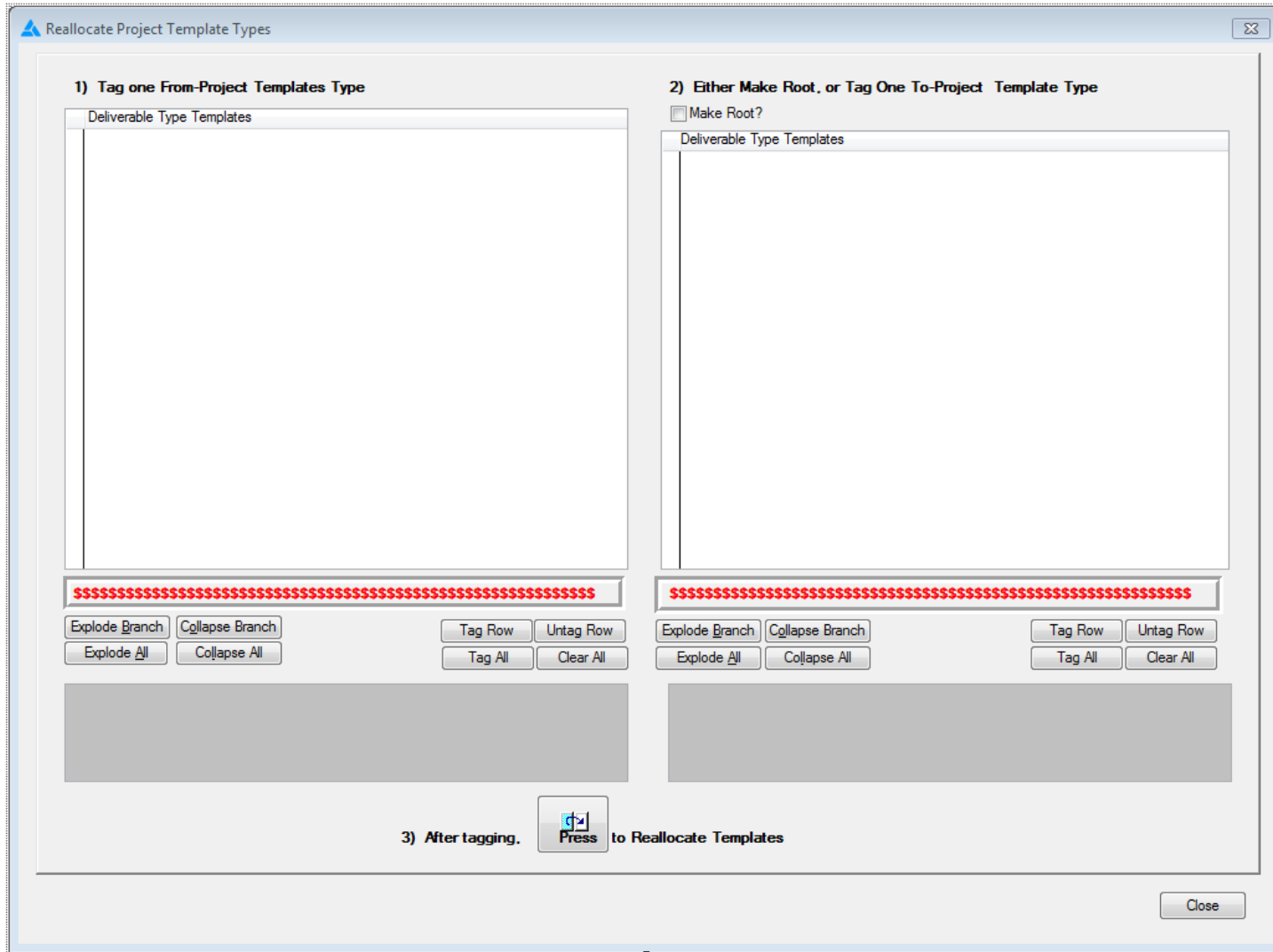
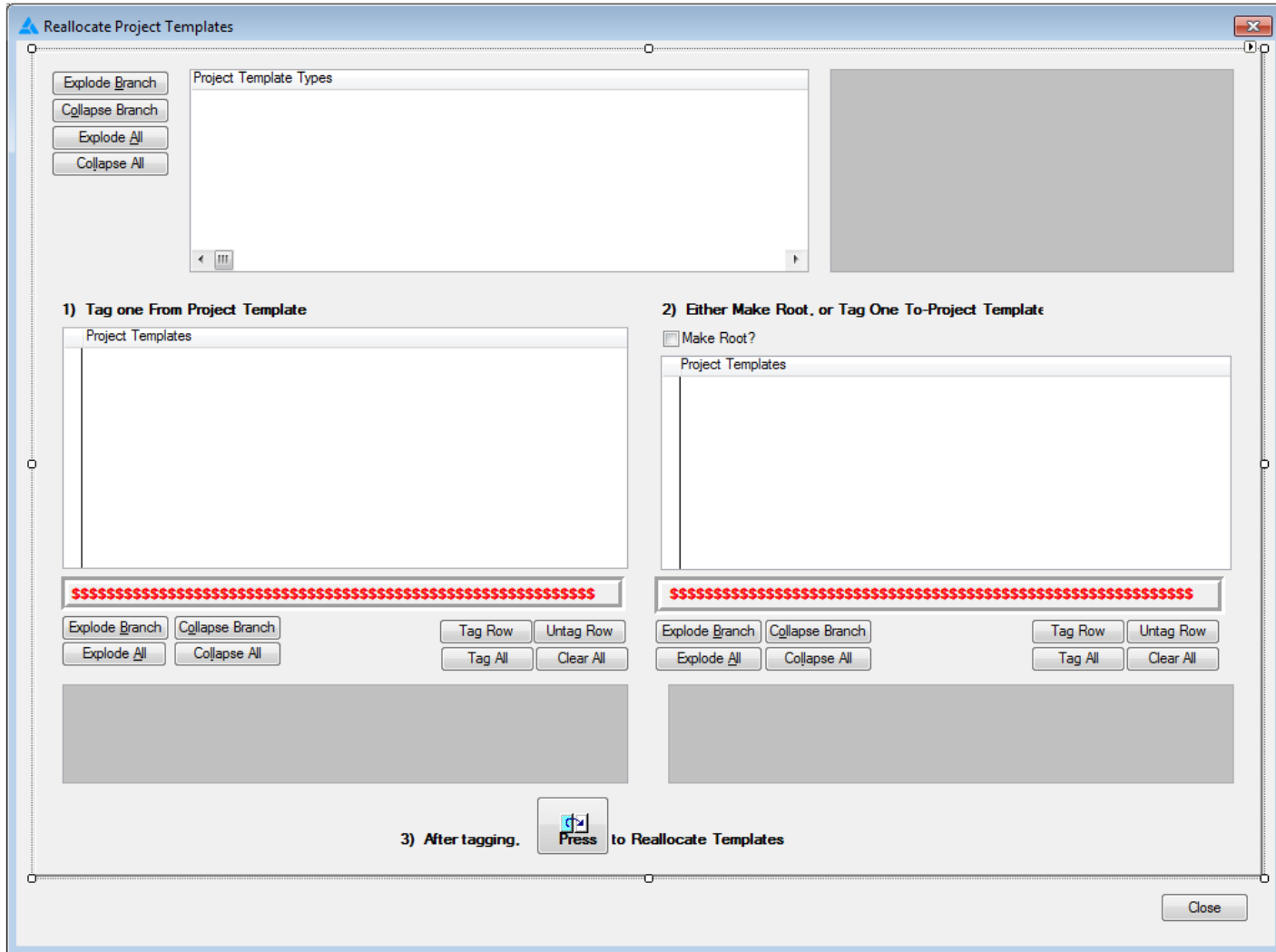


## Generalized Approach to SFR ReAllocation via Equates





---

**Local Data, Generated Declarations, <after Window Structure>**

---

A\_SFRTab        &File  
A2\_SFRTab       &File  
A3\_SFRTab       &File  
A4\_SFRTab       &File  
SFRTab          &File

A\_SFRParentKey &Key  
A\_SFRPkey       &Key  
A\_SFRSeqKey     &Key

A2\_SFRParentKey &Key  
A2\_SFRPkey       &Key  
A2\_SFRSeqKey     &Key

A3\_SFRParentKey &Key  
A3\_SFRPkey       &Key  
A3\_SFRSeqKey     &Key

A4\_SFRParentKey &Key  
A4\_SFRPkey       &Key  
A4\_SFRSeqKey     &Key

SFRPkey         &Key  
SFRSeqKey       &Key

MyFileManager   &FileManager  
A\_MyFileManager &FileManager  
A2\_MyFileManager &FileManager  
A3\_MyFileManager &FileManager  
A4\_MyFileManager &FileManager

---

## Master Procedure Routine Equates

---

### MasterEquates Routine

```
A_SFRTab &= A_ProjectTemplateType
A_SFRParentKey &= A_PTT:ProjectTemplateTypeParentKey
A_SFRPkey &= A_PTT:ProjectTemplateTypePkey
A_SFRSeqKey &= A_PTT:ProjectTemplateTypeSeqKey
A_MyFileManager &= Access:A_ProjectTemplateType

A2_SFRTab &= A2_ProjectTemplateType
A2_SFRParentKey &= A2_PTT:ProjectTemplateTypeParentKey
A2_SFRPkey &= A2_PTT:ProjectTemplateTypePkey
A2_SFRSeqKey &= A2_PTT:ProjectTemplateTypeSeqKey
A2_MyFileManager &= Access:A2_ProjectTemplateType

A3_SFRTab &= A3_ProjectTemplateType
A3_SFRParentKey &= A3_PTT:ProjectTemplateTypeParentKey
A3_SFRPkey &= A3_PTT:ProjectTemplateTypePkey
A3_SFRSeqKey &= A3_PTT:ProjectTemplateTypeSeqKey
A3_MyFileManager &= Access:A3_ProjectTemplateType

A4_SFRTab &= A4_ProjectTemplateType
A4_SFRParentKey &= A4_PTT:ProjectTemplateTypeParentKey
A4_SFRPkey &= A4_PTT:ProjectTemplateTypePkey
A4_SFRSeqKey &= A4_PTT:ProjectTemplateTypeSeqKey
A4_MyFileManager &= Access:A4_ProjectTemplateType

SFRTab &= ProjectTemplateType
SFRPkey &= PTT:ProjectTemplateTypePkey
SFRSeqKey &= PTT:ProjectTemplateTypeSeqKey
MyFileManager &= Access:ProjectTemplateType

!~~~~~
SFRId &= PTT:ProjectTemplateTypeId
SFRParentId &= PTT:ProjectTemplateTypeParentId
SFRSeq &= PTT:ProjectTemplateTypeSeq
```

```

! SFRTId &= xxxxx
SFRSortString &= PTT:sortstring
SFRTabName &= PTT:ProjectTemplateName
!~~~~~
A_SFRId &= A_PTT:ProjectTemplateTypeId
A_SFRParentId &= A_PTT:ProjectTemplateTypeParentId
A_SFRSeq &= A_PTT:ProjectTemplateTypeSeq
! A_SFRTId &= xxxxx
A_SFRSortString &= A_PTT:sortstring
A_SFRTabName &= A_PTT:ProjectTemplateName
!~~~~~
A2_SFRId &= A2_PTT:ProjectTemplateTypeId
A2_SFRParentId &= A2_PTT:ProjectTemplateTypeParentId
A2_SFRSeq &= A2_PTT:ProjectTemplateTypeSeq
! A2_SFRTId &= xxxxx
A2_SFRSortString &= A2_PTT:sortstring
A2_SFRTabName &= A2_PTT:ProjectTemplateName
!~~~~~
A3_SFRId &= A3_PTT:ProjectTemplateTypeId
A3_SFRParentId &= A3_PTT:ProjectTemplateTypeParentId
A3_SFRSeq &= A3_PTT:ProjectTemplateTypeSeq
! A3_SFRTId &= xxxxx
A3_SFRSortString &= A3_PTT:sortstring
A3_SFRTabName &= A3_PTT:ProjectTemplateName
!~~~~~
A4_SFRId &= A4_PTT:ProjectTemplateTypeId
A4_SFRParentId &= A4_PTT:ProjectTemplateTypeParentId
A4_SFRSeq &= A4_PTT:ProjectTemplateTypeSeq
! A4_SFRTId &= xxxxx
A4_SFRSortString &= A4_PTT:sortstring
A4_SFRTabName &= A4_PTT:ProjectTemplateName
!~~~~~

```

---

## Local Variables

---

**SFRId**      **ANY**      **!**

SFRParentId	ANY	!	
SFRSeq	ANY	!	
SFRsortstring	ANY	!	
SFRTabName	ANY	!	
SFRTid	ANY	!	
SFRTypeId	ANY	!	
A2_SFRId	ANY	!	
A2_SFRParentId	ANY	!	
A2_SFRSeq	ANY	!	
A2_SFRsortstring	ANY	!	
A2_SFRTabName	ANY	!	
A2_SFRTid	ANY	!	
A3_SFRId	ANY	!	
A3_SFRParentId	ANY	!	
A3_SFRSeq	ANY	!	
A3_SFRsortstring	ANY	!	
A3_SFRTabName	ANY	!	
A3_SFRTid	ANY	!	
A4_SFRId	ANY	!	
A4_SFRParentId	ANY	!	
A4_SFRSeq	ANY	!	
A4_SFRsortstring	ANY	!	
A4_SFRTabName	ANY	!	
A4_SFRTid	ANY	!	
A_SFRId	ANY	!	
A_SFRParentId	ANY	!	
A_SFRSeq	ANY	!	
A_SFRsortstring	ANY	!	
A_SFRTabName	ANY	!	
A_SFRTid	ANY	!	
AnyDescendentsCheckOK	STRING(3)	!	!
CheckOK	STRING(3)	!	
CurToProcessQueueCount	LONG	!	
CurToProcessQueueIndex	LONG	!	
DescendentAnyNextToProcessCheckOk	STRING(3)	!	!
DisplayString	STRING(255)	!	
ezField	LONG	!	

```

FilesOpened      BYTE           !
LoopParentId    LONG           !
MakeRootYN      STRING(1)     !
MaxSeqNumLV     LONG           !
NewParentId     LONG           !
NextToProcessQueueCount LONG     !
NextToProcessQueueIndex LONG    !
OldParentId     LONG           !
ParentSortStringLV STRING(500) !
PrimePadLV      STRING(500)   !
ProcessDescendantsCheckOk STRING(3) !
ProcessingCheck  STRING(3)    !
StartId         LONG           !
StartParentId   LONG           !
StartSeqNbr     LONG           !
StartTypeId     LONG           !
T_DataModelParentId LONG      !
TagManyMessage  STRING(60)    !
TagManyNewState BYTE         !
TagManyOldState BYTE         !
TagOneMessage   STRING(60)    !
TagOneNewState  BYTE         !
TagOneOldState  BYTE         !
ZeroVal        LONG           !
TagOneQueue     QUEUE,PRE(TOQ) !
TagOneItemID    LONG           !
TagOneItemTypeId LONG        !
TagOneParentId  LONG           !
                END           !
TagManyQueue    QUEUE,PRE(TMQ) !
TagManyItemID   LONG           !
TagManyItemTypeId LONG        !
TagManyParentId LONG           !
                END           !
NextToProcessQueue QUEUE,PRE() !
NxtToProcessQueueId LONG      !
NxtToProcessId  LONG           !

```

```

NxtToProcessParentId LONG      !
NxtToProcessSeq   LONG      !
      END                      !
CurrentToProcessQueue QUEUE,PRE() !
CurToProcessQueueId LONG      !
CurToProcessId   LONG      !
CurToProcessParentId LONG      !
CurToProcessSeq   LONG      !
      END                      !

```

---

**LocalObjectsUltraTree ==> SFRToTree ==> TreeTagManager ==> TakeTagStateChange ==> Code ==> Before ParentCall**

---

```

TagOneOldState = OldState
TagOneNewState = NewState
Do ProcessTagOneStateChange

```

---

**LocalObjectsUltraTree ==> SFRFromTree ==> TreeTagManager ==> TakeTagStateChange ==> Code ==> Before ParentCall**

---

```

TagManyOldState = OldState
TagManyNewState = NewState
Do ProcessTagManyStateChange

```

---

**Control Events ==> ButtonReallocate ==> Accepted ==> after Generated Code:**

---

```

SetCursor(Cursor:Wait)
Beep(Beep:SystemExclamation)
IF makeRootYN = 'Y' THEN
  DO SFRMakeRoot
ELSE
  DO ReallocateSFR
END
MakeRootYN = 'N'
FREE(TagManyQueue)

```



```
FREE(TagOneQueue)
FromTree.RefreshTree
ToTree.RefreshTree
TagMgr_FromTree.UntagAll()
TagMgr_ToTree.UntagAll()
SetCursor()
```

---

Code

---

**ReallocateSFR**    **ROUTINE**

```
Get(TagOneQueue,1)
If TOQ:TagOneItemID = 0 then
  Message('MoveTo not tagged. Canceling move...','ICON:EXCLAMATION)
  Exit
End

Get(TagManyQueue, 1)
If TMQ:TagManyItemID = 0 then  !From-Id
  Message('MoveFrom not tagged. Canceling move...','ICON:EXCLAMATION)
  Exit
End

Do ReAllocationProcess
! Message('Finished Reallocation')
Do ResetSequenceAndSortStrings
```

!~~~~~

**ResetSequenceAndSortStrings Routine**

```
! Message('Starting ResetSequenceAndSortStrings')
A_SFRTid = StartTypeId  !Restricts all Resequencing to just the SFR Type already picked.
A_SFRParentId = 0
A_SFRSeq= 0
StartSeqNbr = 0
Set(A_SFRSeqKey, A_SFRSeqKey)
Loop
  Next(A_SFRTab)
  If ErrorCode() then Break.
  If A_SFRParentId <> 0 Then break.
```

```

StartId = A_SFRId
StartParentId = A_SFRParentId
StartSeqNbr = StartSeqNbr + 1
A_SFRSeq = StartSeqNbr
NewParentId = StartParentId
primepadLV = A_SFRSeq
Do GetParentSortString
Do PrimeandPad
A_SFRSortstring = CLIP(LEFT(ParentSortStringLV)) & CLIP(LEFT(primepadLV))
A_MyFileManager.Update()
IF ERRORCODE() <> 0 THEN
    Message('Reset Error Updating Record Value to Reallocate Hierarchical Position',,ICON:exclamation)
Else
    Do SequenceSFR
END
End

```

!~~~~~

### **ReAllocationProcess Routine**

```

! Message(#0a.. Starting ReAllocationProcess')
SFRId = TMQ:TagManyItemID
GET(SFRTab,SFRPkey)
StartId = SFRId
StartParentId = SFRParentId
StartTypeId = SFRTId
If TMQ:TagManyItemID = TOQ:TagOneItemID then !comparing TagMany From-Id to TagOne To-Id
    Message('MoveFrom and MoveTo the same. No change, exiting',,ICON:EXCLAMATION)
ELSIF SFRParentId = TOQ:TagOneItemID then !comparing From-Id parent to To-Id
    Message('MoveFrom Parent and MoveTo the same. No change, exiting',,ICON:EXCLAMATION)
Else !not the same and parents not the same...move OK
    SFRParentId= TOQ:TagOneItemID !actually just changing the TagManyParentID to the TagOneParentId...
    NewParentId = SFRParentId
    Do GetMaxSeqNum
    SFRSeq= maxseqnumLV
    primepadLV = SFRSeq
    Do GetParentSortString
    Do PrimeandPad

```

```

SFRsortstring = CLIP(LEFT(ParentSortStringLV)) & CLIP(LEFT(primepadLV))
MyFileManager.Update()
IF ERRORCODE() <> 0 THEN
    Message('ReAllocate Error Updating Record Value to Reallocate Hierarchical Position',,ICON:exclamation)
Else
    Do SequenceSFR
    If StartParentId <> 0 then
        StartId = NewParentId !Why shouldn't this be NewParentId
        Do SequenceSFR
    Else
        Exit
    End
End
End

```

```

SFRMakeRoot    ROUTINE
!Message('Starting SFRMakeRoot')
Get(TagManyQueue, 1)
A_SFRIId = TMQ:TagManyItemID
Get(A_SFRTab, A_SFRPkey)
If A_SFRParentId = 0 then
    Message('This item already is at root level...exiting',,ICON:EXCLAMATION)
    Exit
Else
    Do MakeRoot
End

```

```

MakeRoot    Routine
!Message('Starting Make Root')
StartTypeId = A_SFRTId
NewParentID = 0
Do GetMaxSeqNum
primepadLV = maxseqnumLV
Do PrimeandPad
A_SFRSeq = maxseqnumLV
A_SFRParentId = NewParentID
A_SFRSortString = CLIP(LEFT(primepadLV))

```

```

If A_MyFileManager.UPDATE()<>Level:Benign then
  Message('Could not make root. Resetting...')
  MakeRootYN = 'N' !Resetting MakeRoot Flag
  Exit
Else
  MakeRootYN = 'N' !Resetting MakeRoot Flag
  StartId = A_SFRId
  Do SequenceSFR
End

```

### **SequenceSFR**      **Routine**

```

! Message('Starting SequenceSFR')
Free(NextToProcessQueue)
Free(CurrentToProcessQueue)
A4_SFRId = StartId
Get(A4_SFRTab, A4_SFRPkey)
Do Check
If CheckOK = 'No' then
  ProcessingCheck = 'No'
  Exit
Else
  ProcessingCheck = 'Yes'
  Exit
End

```

### **Check**            **Routine**

```

!Message('Starting Check')
A4_SFRParentId = StartID
Get(A4_SFRTab, A4_SFRParentKey)
If ErrorCode() = 35 then
  CheckOK = 'Yes'
  Exit
Else
  LoopParentId = StartId
  Do AnyDescendents
  If AnyDescendentsCheckOK = 'No' then
    CheckOk = 'No'

```

```

Exit
Else
  CheckOk = 'Yes'
  Do DescendentAnyNextToProcess
  If DescendentAnyNextToProcessCheckOK = 'No' then
    CheckOk = 'No'
    Exit
  Else
    CheckOk = 'Yes'
  End
End
End
End

```

#### **AnyDescendents Routine**

```

!Message('Starting DoesHaveDescendents')
A4_SFRParentId = LoopParentId
Get(A4_SFRTab, A4_SFRParentKey)
If ErrorCode() = 35 then
  AnyDescendentsCheckOK = 'Yes'
  Exit
Else
  AnyDescendentsCheckOK = 'No'
  Set(A4_SFRParentKey, A4_SFRParentKey)
  Loop
    Next(A4_SFRTab)
    If ErrorCode() = 33 then Break.
    If A4_SFRParentId <> LoopParentId then Break.
    AnyDescendentsCheckOK = 'Yes'
    NxtToProcessQueueId = NxtToProcessQueueId +1
    NxtToProcessId = A4_SFRId
    NxtToProcessParentId = A4_SFRParentId
    NxtToProcessSeq = A4_SFRSeq
    Add(NxtToProcessQueue)
  End
End
End

```

#### **DescendentAnyNextToProcess Routine**

```

!Message('Starting DescendentAnyNextToProcess')
NextToProcessQueueCount = Records(NextToProcessQueue)
If NextToProcessQueueCount > 0 then
  Do ProcessDescendents
  If ProcessDescendentsCheckOk = 'No' then
    DescendentAnyNextToProcessCheckOk = 'No'
    Exit
  Else
    DescendentAnyNextToProcessCheckOk = 'Yes'
    Exit
  End
End

```

### ProcessDescendents Routine

```

! Message('Starting ProcessDescendents')
Do TransferNextToCurrent
Sort(CurrentToProcessQueue, CurToProcessParentId, CurToProcessSeq)
CurToProcessQueueCount = Records(CurrentToProcessQueue)
CurToProcessQueueIndex = 0
Loop CurToProcessQueueCount Times
  CurToProcessQueueIndex = CurToProcessQueueIndex + 1
  Get(CurrentToProcessQueue, CurToProcessQueueIndex)
  A4_SFRIId= CurToProcessId
  Get(A4_SFRTab, A4_SFRPkey)
  T_DataModelParentId = CurToProcessParentId
!~~~~~
  A2_SFRIId= CurToProcessParentId
  Get(A2_SFRTab, A2_SFRPkey)
!~~~~~
  A4_SFRParentId = CurToProcessParentId !actually just changing the parent...
  If OldParentId <> CurToProcessParentId Then !For Simple and Typed version
    MaxSeqNumLv = 0 !For Simple and Typed version
    OldParentId = CurToProcessParentId !For Simple and Typed version
  End !For Simple and Typed version
  NewParentId = A4_SFRParentId
  MaxseqnumLV = MaxseqnumLV + 1 ! Setting value for Sequence Number within a Level
  A4_SFRSeq= maxseqnumLV ! Setting Sequence Number for SFR level

```

```

primepadLV = A4_SFRSeq
Do GetParentSortString
Do PrimeandPad
A4_SFRSeq = maxseqnumLV * 1000 ! Setting Sequence Number for SFR level * 1000 to avoid Unique Errors until update complete
A4_SFRSortString = CLIP(LEFT(ParentSortStringLV)) & CLIP(LEFT(primepadLV))
A4_MyFileManager.Update()
ProcessDescendentsCheckOk = 'Yes'
LoopParentId = CurToProcessId !sets to then go down a level
Do AnyDescendents
End
Do ReCastSequenceNumbers
Do DescendentAnyNextToProcess

```

#### **TransferNextToCurrent**    **Routine**

```

!Message('Starting TransferNextToCurrent')
MaxseqnumLV = 0 !Recycling value to pick up next level of sequence numbers
Free(CurrentToProcessQueue)
NextToProcessQueueIndex = 0
Loop NextToProcessQueueCount Times
    NextToProcessQueueIndex = NextToProcessQueueIndex +1
    Get(NextToProcessQueue, NextToProcessQueueIndex)
    CurToProcessQueueId = NxtToProcessQueueId
    CurToProcessId = NxtToProcessId
    CurToProcessParentId = NxtToProcessParentId
    CurToProcessSeq = NxtToProcessSeq
    Add(CurrentToProcessQueue)
End
Free(NextToProcessQueue)

```

#### **ReCastSequenceNumbers**    **Routine** !This resets the sequence numbers by dividing those just set by 1000

```

! Message('Starting ReCastSequenceNumbers')
Sort(CurrentToProcessQueue, CurToProcessParentId, CurToProcessSeq)
CurToProcessQueueCount = Records(CurrentToProcessQueue)
CurToProcessQueueIndex = 0
Loop CurToProcessQueueCount Times
    CurToProcessQueueIndex = CurToProcessQueueIndex +1

```

```

    Get(CurrentToProcessQueue, CurToProcessQueueIndex)
    A4_SFRIId= CurToProcessId
    Get(A4_SFRTab, A4_SFRPkey)
    A4_SFRSeq = A4_SFRSeq/1000 ! Setting Sequence Number for SFR level
    A4_MyFileManager.Update()
End

```

#### GetParentSortString Routine

```

IF NewParentId <> 0 then
    A2_SFRIId = NewParentId
    Get(A2_SFRTab, A2_SFRPkey)
    ParentSortStringLV = Clip(Left(A2_SFRSortString))
Else
    ParentSortStringLV = ' '
End

```

#### GetMaxSeqNum ROUTINE

```

!Message('Starting GetMaxSeqNum')
A3_SFRTId = StartTypeId
A3_SFRParentId = NewParentId
A3_SFRSeq= 0
SET(A3_SFRSeqKey, A3_SFRSeqKey)
LOOP
    NEXT(A3_SFRTab)
    If ERRORCODE() <> 0 Then Break.
    If A3_SFRTId <> StartTypeId Then Break.
    IF A3_SFRParentId <> NewParentId Then Break.
    MaxseqnumLV = A3_SFRSeq
END
MaxseqnumLV = maxseqnumLV + 1

```

#### PrimeAndPad ROUTINE

```

DATA
SortString STRING(4)
CODE
SortString = primepadLV
LOOP

```



```

IF LEN( CLIP(LEFT(SortString)) ) % 4 = 0 THEN ! if modulus of dividing the sortstring by 4 is not empty, then keep padding...
  BREAK
END
sortstring = '0' & CLIP(LEFT(SortString))
END
primepadLV = CLIP(LEFT(SortString))

```

#### **ProcessTagManyStateChange Routine**

```

If TagManyNewState = 1 then !1=Tagged..from not tagged to tagged
  Do AddTagToTagManyQueue
ElseIf TagManyNewState = 0 then !0 = Not Tagged... from tagged to not tagged
  Do RemoveTagFromTagManyQueue
Else !2... this in not valid to be tagged.
  Exit
End

```

#### **AddTagToTagManyQueue Routine**

```

TMQ:TagManyItemId = SFRId
TMQ:TagManyItemTypeId = SFRTId
TMQ:TagManyParentId = SFRParentId
TagManyMessage = 'Accepted: Tagging Ok'
ADD(TagManyQueue,1)
Display

```

#### **RemoveTagFromTagManyQueue Routine**

```

Get(TagManyQueue, 1)
Delete(TagManyQueue)
TagMgr_ToTree.UntagRow()
TagManyMessage = 'Tag Removed'
Display

```

#### **ResetTagMany Routine**

```

TagMgr_ToTree.UntagAll()

```

#### **ProcessTagOneStateChange Routine**

```

If TagOneNewState = 1 then !1=Tagged..from not-tagged to tagged
  TagOneMessage = 'Accepted: Tagging Ok'

```

```
Display
Do AddTagToTagOneQueue
ElsIf TagOneNewState = 0 then !0 = Not Tagged... from tagged to not-tagged
Do RemoveTagFromTagOneQueue
Else !2... this in not valid to be tagged.
Exit
End
```

#### **AddTagToTagOneQueue Routine**

```
TOQ:TagOneItemId = A2_SFRId
TOQ:TagOneItemTypeId = A2_SFRTId
TOQ:TagOneParentId = A2_SFRParentId
ADD(TagOneQueue, 1)
TagOneMessage = 'Accepted: Tagging Ok'
```

#### **RemoveTagFromTagOneQueue Routine**

```
Get(TagOneQueue, 1)
Delete(TagOneQueue)
TagMgr_FromTree.UntagRow()
TagOneMessage = 'Tag Removed'
Display
```

#### **ResetTagOne Routine**

```
TagMgr_FromTree.UntagAll()
```