**LA7SMB**   ;DALOI/JMC - Shipping Manifest Build ;11/16/11 13:08  
      ;;5.2;AUTOMATED LAB INSTRUMENTS;\*\*27,46,64,74\*\*;Sep 27, 1994;Build 229  
      ;  
**EN**   ;  
      ;  
      D CLEANUP  
      S LA7QUIT=0  
      ;  
      ; Select shipping configuration  
      I '$D(^LAHM(62.9,"C",DUZ(2))) D Q ;  
      . W !,$C(7),"No configurations for your current division."  
      . H 2  
      S LA7SCFG=$$SSCFG^LA7SUTL(1)  
      I LA7SCFG<1 D CLEANUP Q  
      ;  
      ; Determine if there's an active manifest.  
      S LA7SM=$$CHKSM^LA7SMU(+LA7SCFG)  
      I LA7SM<0 D Q  
      . D EN^DDIOL($P(LA7SM,"^",2),"","!?5")  
      . D CLEANUP  
      ;  
      I LA7SM=0 D  
      . N DIR,DIRUT,DTOUT,X,Y  
      . S DIR(0)="YO",DIR("A",1)="There's no open shipping manifest for "\_$P(LA7SCFG,"^",2)  
**. S DIR("A")="Do you want to start one",DIR("B")="YES" ;;\*mod gwb/chs 2-27-13 original mod 1-17-08 change from NO to YES**  
      . D ^DIR  
      . I Y'=1 S LA7QUIT=1 Q  
      . S LA7SM=$$CSM^LA7SMU(+LA7SCFG)  
      . I LA7SM<1 D EN^DDIOL($P(LA7SM,"^",2),"","!?5") S LA7QUIT=1  
      ;  
      ; Only starting a new manifest, no building  
      I $G(LA7SMON) Q  
      ;  
      I LA7QUIT=1 D CLEANUP Q  
      ;  
      D ADATE^LA7SMU1  
      I LA7QUIT=1 D CLEANUP Q  
      ;  
      ; Flag to exclude previously removed tests from building.  
      S LA7EXPRV=$$ASKPREV^LA7SMU1(+LA7SCFG)  
      I LA7EXPRV<0 S LA7QUIT=1  
      ;  
      I LA7QUIT=1 D CLEANUP Q  
      ;  
**DQ**   ; Taskman entry point  
      ; Build list of tests and criteria for manifest.  
      S LA7SCFG(0)=$G(^LAHM(62.9,+LA7SCFG,0))  
      I '$D(ZTQUEUED) D EN^DDIOL("Using shipping manifest# "\_$P(LA7SM,"^",2),"","!?5")  
      ;  
      ; Lock this shipping manifest  
      D LOCK^DILF("^LAHM(62.8,+LA7SM,0)")  
      I '$T D Q  
      . I '$D(ZTQUEUED) D EN^DDIOL("Unable to obtain lock for shipping manifest "\_$P(LA7SCFG,"^",2),"","!?5")  
      . D CLEANUP  
      ;  
      ; Update status  
      D SMSUP^LA7SMU(LA7SM,2,"SM03")  
      S LA7SMCNT=0  
      ;  
      ; Build TMP global with test profiles  
      D SCBLD^LA7SM1(+LA7SCFG)  
      ;  
      S LA7AA=""  
      F S LA7AA=$O(^TMP("LA7SMB",$J,LA7AA)) Q:LA7AA="" D  
      . N LA7END,LRSS  
      . I '$D(ZTQUEUED) D EN^DDIOL("Searching accession area: "\_$P($G(^LRO(68,LA7AA,0)),"^"),"","!?5")  
      . ; Use selected accession date else get current accession day for this accession area  
      . I $G(LA7AA(LA7AA)) S LA7AD=$P(LA7AA(LA7AA),"^")  
      . E S LA7AD=$$AD^LA7SUTL(LA7AA)  
      . S LRSS=$P($G(^LRO(68,LA7AA,0)),"^",2)  
      . S LA7AN=+$P($G(LA7AA(LA7AA)),"^",2),LA7LAN=+$P($G(LA7AA(LA7AA)),"^",3),LA7END=0  
      . I LA7AN S LA7AN=LA7AN-1  
      . F S LA7AN=$O(^LRO(68,LA7AA,1,LA7AD,1,LA7AN)) Q:'LA7AN!(LA7END) D SCAN  
      ;  
      ; Update status  
      D SMSUP^LA7SMU(LA7SM,1,"SM02")  
      ;  
      ; Release lock on this shipping manifest  
      L -^LAHM(62.8,+LA7SM,0)  
      ;  
      I '$D(ZTQUEUED) D  
      . N DIR,DIRUT,DIROUT,DTOUT,X,Y  
      . D EN^DDIOL("There were "\_$S(LA7SMCNT:LA7SMCNT,1:"NO")\_" specimens added","","!?5")  
      . D ASK^LA7SMP(LA7SM)  
      D CLEANUP  
      Q  
      ;  
**SMONLY**   ; Start a shipping manifest only, no building  
      ;  
      N LA7SMON  
      S LA7SMON=1  
      D EN  
      I $G(LA7SCFG),$G(LA7SM)>0 D EN^DDIOL("Shipping manifest# "\_$P(LA7SM,"^",2)\_" is available","","!?5")  
      D CLEANUP  
      Q  
      ;  
      ;  
**SCAN**   ; Scan accession for tests to build  
      ;  
      N LA762,LA76805,LA7DIV,LA7END,X  
      ;  
      I LA7LAN,LA7AN>LA7LAN S LA7END=1 Q  
      ;  
      ; Don't build controls  
      I $P($G(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,0)),"^",2)=62.3 Q  
      ;  
      ; Don't build uncollected specimens  
      I '$P(LA7SCFG(0),"^",14),'$P($G(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,3)),"^",3) Q  
      ;  
      ; Don't build if configuration has specific ordering locations and accession's ordering location is not on list.  
      I $$CHKOLOC^LA7SM1(LA7AA,LA7AD,LA7AN,+LA7SCFG)<1 Q  
      ;  
      ; Get Specimen type - if no specimen then quit  
      S (LA762,LA76805)=0  
      S X=+$O(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,5,0))  
      I X D  
      . S X=$G(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,5,X,0))  
      . S LA76805=+$P(X,"^"),LA762=+$P(X,"^",2)  
      I LA76805<1 Q  
      ;  
      ; Accession's division  
      S LA7DIV=+$P($G(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,.4)),"^")  
      ;  
      S LA7UID=$P($G(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,.3)),"^")  
      I LA7UID="" S LA7UID=$$LRUID^LRX(LA7AA,LA7AD,LA7AN)  
      S LA760=0  
      F S LA760=$O(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,4,LA760)) Q:'LA760 D  
      . ; Not looking for this test.  
      . I '$D(^TMP("LA7SMB",$J,LA7AA,LA760)) Q  
      . ; Set lock.  
      . D LOCK68  
      . I '$T D Q  
      . . I $D(ZTQUEUED) Q  
      . . D EN^DDIOL("Unable to obtain lock for accession "\_LA7UID\_" test "\_$$GET1^DIQ(60,LA760\_",",.01),"","!?5")  
      . ; NOTE \*\*\* Do NOT add any "QUIT" after this point unless releasing LOCK set above \*\*\*.  
      . ; Test's zeroth node.  
      . S LA760(0)=$G(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,4,LA760,0))  
      . ; Test completed - skip  
      . I "CY^EM^SP"'[LRSS,$P(LA760(0),"^",5) D UNLOCK68 Q  
      . ; Test already on shipping manifest - skip  
      . I $$SHIPCK^LA7SMU1(LA7UID,LA7AA,LA760,$P(LA760(0),"^",10)) D UNLOCK68 Q  
      . ; Previously removed - skip  
      . I LA7EXPRV,$$PREV^LA7SMU1(LA7UID,$P(LA760(0),"^")) D UNLOCK68 Q  
      . ; Test urgency  
      . S LA76205=+$P(LA760(0),"^",2)  
      . I LA76205>49 S LA76205=$S(LA76205=50:9,1:LA76205-50)  
      . ; Check if test is eligible for manifest  
      . D SCHK^LA7SM1  
      . I LA7FLAG S LA7FLAG=$$CKTEST(LA7AA,LA7AD,LA7AN,LA760)  
      . ; Add test to shipping manifest.  
      . I LA7FLAG D ADD^LA7SMB1  
      . ; Release lock.  
      . D UNLOCK68  
      Q  
      ;  
      ;  
**CKTEST(LA7AA,LA7AD,LA7AN,LA760)**   ; Check other tests on accession if test is part of another panel that  
      ; has been flagged for shipping.  
      ; Call with LA7AA = ien of accession area.  
      ; LA7AD = accession date  
      ; LA7AN = accession number  
      ; LA760 = ien of lab test  
      ; Returns LA7FLAG = 0 (part of another panel)  
      ; = 1 (not part of another panel)  
      ;  
      N LA7FLAG,LA7PCNT,LA7K,LA7J,X  
      ;  
      K ^TMP("LA7TREE",$J)  
      ;  
      S LA7FLAG=1  
      S LA7AD(LA7AD)=""  
      S LA7K=+$P($G(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,0)),"^",3)  
      ;  
      I LA7K D  
      . ; Check original accession date.  
      . S LA7AD(LA7K)=""  
      . ; Check rollover accession  
      . I $P($G(^LRO(68,LA7AA,1,LA7K,1,LA7AN,9)),"^") S LA7AD($P($G(^LRO(68,LA7AA,1,LA7K,1,LA7AN,9)),"^"))=""  
      S LA7AD=0  
      F S LA7AD=$O(LA7AD(LA7AD)) Q:'LA7AD D  
      . S LA7J=0  
      . F S LA7J=$O(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,4,LA7J)) Q:'LA7J D  
      . . I LA7J=LA760 Q  
      . . ; Not on manifest  
      . . I '$P($G(^LRO(68,LA7AA,1,LA7AD,1,LA7AN,4,LA7J,0)),"^",10) Q  
      . . S LA7PCNT=0 D UNWIND(LA7J)  
      ;  
      ; Test is part of another test previously shipped.  
      I $D(^TMP("LA7TREE",$J,LA760)) S LA7FLAG=0  
      ;  
      K ^TMP("LA7TREE",$J)  
      ;  
      Q LA7FLAG  
      ;  
**UNWIND(LA760)**   ; Unwind profile - set tests into global ^TMP("LA7TREE",$J).  
      ; Initialize variable LA7PCNT=0 before calling.  
      ; Kill ^TMP("LA7TREE",$J) before calling.  
      ;  
      N I,II  
      ;  
      ; Recursive panel, caught in a loop.  
      I $G(LA7PCNT)>50 Q  
      ; Test does not exist in file 60.  
      I '$D(^LAB(60,LA760,0)) Q  
      ; Bypass "workload" type tests.  
      I $P(^LAB(60,LA760,0),"^",4)="WK" Q  
      ; Atomic test  
      I $P(^LAB(60,LA760,0),"^",5)'="" S ^TMP("LA7TREE",$J,LA760)="" Q  
      ; Check panels  
      I $O(^LAB(60,LA760,2,0)) D  
      . ; Increment panel counter.  
      . S LA7PCNT=$G(LA7PCNT)+1  
      . S I=0  
      . ; Expand test on panel.  
      . F S I=$O(^LAB(60,LA760,2,I)) Q:'I D  
      . . ; IEN of test on panel.  
      . . S II=+$G(^LAB(60,LA760,2,I,0))  
      . . ; Recursive panel, panel calls itself.  
      . . I II,II=LA760 Q  
      . . I II S ^TMP("LA7TREE",$J,LA760)="" D UNWIND(II)  
      Q  
      ;  
**LOCK68**   ; Lock entry in file 68  
      ; Called from above, LA7SM  
      ;  
      D LOCK^DILF("^LRO(68,LA7AA,1,LA7AD,1,LA7AN,4,LA760)")  
      ;  
      Q  
      ;  
**UNLOCK68**   ; Unlock entry in file 68  
      ; Called from above, LA7SM  
      ;  
      L -^LRO(68,LA7AA,1,LA7AD,1,LA7AN,4,LA760) ; Release lock.  
      ;  
      Q  
      ;  
**CLEANUP**   ; Cleanup variables  
      ;  
      I $D(ZTQUEUED) S ZTREQ="@"  
      ;  
      F LA7I="LA7SMB","LA7SMB-R","LA7SMADD","LA7TREE" K ^TMP(LA7I,$J)  
      ;  
      K LA760,LA76205,LA76805,LA7AA,LA7AD,LA7AN,LA7DATA,LA7EXPRV,LA7FLAG,LA7I,LA7LAN,LA7PCNT,LA7QUIT,LA7SCFG,LA7SM,LA7SMCNT,LA7UID,LA7X  
      K LRDFN  
      Q