Option Explicit

Const ReturnNoSolution As Integer = -32767

Const ReturnNotMovableDelta As Integer = -1000

Dim RecurLevel As Integer

Dim ListRecurCall As String

'------------------------------------------------------------ AU\_NPM\_MarginManageProtectedFlights

' modify the FDATime

' lprioModel must be assigned to GPrioModel\_TimeMode\_OnSchedule or GPrioModel\_TimeMode\_OnMargin

' in this case we use the schedule or the margin to make the calculation

' lAllFl is list off flights to find Margin solution (All except Pflights and ExplicitB flights)

' when finding margin solution, if default priority value is baseline, prio is put to lowest prio

' If Time not after is greater then the Hotspot end, initialize to the end of the hotspot

' BE CAREFULL a B priority on margins flights exclude flight from management

Function AU\_NPM\_ManageMarginPrioFlights\_Main(lprioModel As Integer, \_

lAll\_AUFlights As CL\_AllFlights, lMyFlightsIx As CL\_AUFlightsIx, \_

ByRef lAllFl() As Integer, lAllFl\_nb As Integer, \_

ByRef lAddFlight() As Integer, ByRef lMarginflight\_Nb As Integer) As Integer

Dim li As Integer

Dim lj As Integer

Dim lName As String

Dim ltime As Date

'Dim lAllFl\_prio() As Integer ' containt prio off All flight for this AU (Ix in Allflight)

'Dim lAllFlSortedOnBaseline() As Integer ' containt All flight to manage the margin sorted

Dim lMarginFl() As Integer ' containt All flight twho have a margin

Dim lMarginSortedFl() As Integer ' containt All flight to manage the margin sorted

Dim lMarginFl\_nb As Integer ' nb of flight to manage the margin

Dim lPrioOnlyFl() As Integer ' containt All flight twho have a margin

Dim lPrioOnlySortedFl() As Integer ' containt All flight to manage the margin sorted

Dim lPrioOnlyFl\_nb As Integer ' nb of flight to manage the margin

Call EX\_Mess(EX\_MESS\_Start, "AU NPRIO MARGINandPrio: " & lAll\_AUFlights.AUName)

'ListRecurCall = ""

' do nothing if only 1 flight

If lAllFl\_nb < 2 Then

Exit Function

End If

Dim lMargeIx As Integer

' slot management

Dim lMySlotsValue() As Date

Dim lMySlotsValueSorted() As Date

Dim lMySlotsUsed() As Integer

' get the flight with margin to manage

' get the list of flight with margins to manage

lMarginFl\_nb = 0

lPrioOnlyFl\_nb = 0

' get margin flights and prio only flights

' BE CAREFULL the management of explicit B on margin flight

' explicit B normally not part of the flights

If lAllFl\_nb > 0 Then

ReDim lMarginFl(lAllFl\_nb)

ReDim lMySlotsValue(lAllFl\_nb)

ReDim lPrioOnlyFl(lAllFl\_nb)

ReDim lMySlotsUsed(lAllFl\_nb)

For li = 0 To lAllFl\_nb - 1

lMySlotsValue(li) = lAll\_AUFlights.GetFDATime(lAllFl(li))

lMySlotsUsed(li) = -1

If (lAll\_AUFlights.GetMarginNotAfterTimeIsInit(lAllFl(li)) = True) And \_

(lAll\_AUFlights.GetPrio(lAllFl(li)) <> GPrioSuspended) Then

lMarginFl(lMarginFl\_nb) = lAllFl(li)

lMarginFl\_nb = lMarginFl\_nb + 1

Else

lPrioOnlyFl(lPrioOnlyFl\_nb) = lAllFl(li)

lPrioOnlyFl\_nb = lPrioOnlyFl\_nb + 1

End If

Next li

' get my slots and manage the list of margin or prio flights

ReDim lMySlotsValueSorted(lAllFl\_nb)

' Sort my Time slots (by FDATime)

Call AU\_NPS\_SortATimeTable(lMySlotsValue, lMySlotsValueSorted, lAllFl\_nb)

Erase lMySlotsValue

If lMarginFl\_nb < 1 Then

Erase lMarginFl

Else

' Sort my Margin flights by prio and Margins and Schedule

ReDim lMarginSortedFl(lMarginFl\_nb)

Call AU\_NPS\_SortByPrioAndMarginTimeNotAfterAndBaselineTime(lAll\_AUFlights, lMarginFl, lMarginSortedFl, lMarginFl\_nb)

Erase lMarginFl

' loop on earch Margin flights

' put Margin flights on available slot

Call EX\_Log\_Init

For lMargeIx = 0 To lMarginFl\_nb - 1

' for test

Dim lCallsign As String

Dim lMarginTime As Date

Dim lMarginFlightIx As Integer

lMarginFlightIx = lMarginSortedFl(lMargeIx)

lCallsign = lAll\_AUFlights.GetCallsignICAO(lMarginFlightIx)

Dim lSlotAssigned As Integer

If lMargeIx = 73 Then

lMargeIx = lMargeIx

End If

lMarginTime = lAll\_AUFlights.GetMarginNotAfterTime(lMarginFlightIx)

lSlotAssigned = AU\_NPM\_ManageMarginPrioFlights\_ManageTimeSolution(lAll\_AUFlights, \_

lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb, \_

lMarginTime, lMarginFlightIx)

' Test if there is a slot

If lSlotAssigned < 0 Then

' no time solution

Call AU\_NPM\_MsgboxStop(" ERROR TO ASSIGN Margin flight to a slot : " & lCallsign, \_

lAll\_AUFlights, lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb)

End If

Next lMargeIx

End If

' manage the other type of flights

If lPrioOnlyFl\_nb < 1 Then

'Erase lMySlotsValue

Erase lPrioOnlyFl

'Erase lMySlotsUsed

Else

' AU slot has been assigned to Margin flights

' now manage the other flights (prio only)

ReDim lPrioOnlySortedFl(lPrioOnlyFl\_nb)

Call AU\_NPS\_SortByPrioAndSchedule(lAll\_AUFlights, lPrioOnlyFl, lPrioOnlySortedFl, lPrioOnlyFl\_nb)

Erase lPrioOnlyFl

' assign prio flights be carrefull baseline flights must be assign first then flight with number

Call AU\_NPM\_ManageMarginPrioFlights\_AssignOtherFlights(lAll\_AUFlights, \_

lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb, \_

lPrioOnlySortedFl(), lPrioOnlyFl\_nb)

Erase lMarginSortedFl

End If

' return the nb of margin flights

lMarginflight\_Nb = lMarginFl\_nb

End If

' at the en Pack the flights on slots (use the available AU slot

' needed if there is some Suspended flights

If lMarginFl\_nb > 0 Or lPrioOnlyFl\_nb > 0 Then

Call AU\_NPM\_ManageMarginPrioFlights\_UseAvailableSlots(lAll\_AUFlights, \_

lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb)

End If

' assign the flights to the slots

'------ Update the FDA time ------

' First assigne the FDA time on margins flights

For li = 0 To lAllFl\_nb - 1

ltime = lMySlotsValueSorted(li)

' if the slot is used by a margin flight assign it

If lMySlotsUsed(li) <> -1 Then

' assign the Margin flights

lName = lAll\_AUFlights.GetCallsignICAO(lMySlotsUsed(li))

Call lAll\_AUFlights.SetFDATime(lMySlotsUsed(li), lMySlotsValueSorted(li))

End If

Next li

' ---------- Manage the Suspended flights

If lPrioOnlyFl\_nb > 0 Then

' Assign it the hotspot end time in fdaValue

Call AU\_NPM\_ManageMarginPrioFlights\_AssignSuspendedFlights(lAll\_AUFlights, \_

lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb, \_

lPrioOnlySortedFl(), lPrioOnlyFl\_nb)

End If

Erase lPrioOnlySortedFl

Erase lMySlotsValueSorted

Erase lMySlotsUsed

' return nb of impacted flights to add on next list

AU\_NPM\_ManageMarginPrioFlights\_Main = 0

Call EX\_Mess(EX\_MESS\_End, "AU NPRIO MARGINandPrio: " & lAll\_AUFlights.AUName)

End Function

'----------------------------------------- AU\_NPM\_ManageMarginPrioFlights\_ManageTimeSolution

' try and manage the solution before the target time

' At this stage the target slot is already used by another flight

' first look if there is an available slot before the target one to shift earlier the others

' if yes :

' try to shift the flights to make a hole at the target slot

' otherwise give a slot before for this flight

' if no slot return -1

Function AU\_NPM\_ManageMarginPrioFlights\_ManageTimeSolution( \_

lAll\_AUFlights As CL\_AllFlights, \_

ByRef lMySlotsValueSorted() As Date, ByRef lMySlotsUsed() As Integer, lAllFl\_nb As Integer, \_

lTargetTime As Date, \_

lFlightIx As Integer) As Integer

'Dim lSlotEarlierPossible As Integer

Dim lAvailableSlot\_Earlier As Integer

Dim lAvailableSlot\_Later As Integer

Dim lEarliestTime As Date

Dim lTargetSlot As Integer

Dim lReturn As Integer

'lAvailableSlot\_Earlier = -1

RecurLevel = 0

'Call EX\_Log\_Init

' earliest time of the margin flight

lEarliestTime = lAll\_AUFlights.GetRefBlockTime(lFlightIx) \_

- GHspt\_FlightEarlyDeparture\_forDate

lTargetSlot = AU\_NPMF\_GetTargetSlots(lMySlotsValueSorted, lAllFl\_nb, lTargetTime, lEarliestTime)

' Test if there is a slot

If lTargetSlot < 0 Then

' no time solution

Call AU\_NPM\_MsgboxStop(" ERROR TO DO SOMETHING Margin flight with no time solution : " & lAll\_AUFlights.GetCallsignICAO(lFlightIx), \_

lAll\_AUFlights, lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb)

Call EX\_Log(RecurLevel, " --> No slot solution for " & lFlightIx & " on targetTime " & lTargetTime)

lReturn = -1

Else

' try manage solution by shifting flights earlier first

' the slot could ber later then the one asked

lAvailableSlot\_Earlier = AU\_NPM\_ManageMarginPrioFlights\_ManageSolutionEarlier(lAll\_AUFlights, \_

lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb, \_

lTargetSlot, lFlightIx)

If lAvailableSlot\_Earlier > -1 Then

' Found a place in the slot list earlierassign it

lMySlotsUsed(lAvailableSlot\_Earlier) = lFlightIx

'ListRecurCall = ListRecurCall & " S= " & lAvailableSlot\_Earlier & " |"

Call EX\_Log(RecurLevel, "End FL: " & lFlightIx & " EARLIER Solution is slot: " & lAvailableSlot\_Earlier)

lReturn = lAvailableSlot\_Earlier

Else

' No possible slot earlier

' test and get if there is available slot later

lAvailableSlot\_Later = AU\_NPMF\_GetLaterAvailableSlots(lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb, \_

lTargetSlot + 1, lEarliestTime)

If lAvailableSlot\_Later > -1 Then

' put it at this place

lMySlotsUsed(lAvailableSlot\_Later) = lFlightIx

Call EX\_Log(RecurLevel, "End FL: " & lFlightIx & " LATER Solution is slot: " & lAvailableSlot\_Later)

lReturn = lAvailableSlot\_Later

Else

' in this case no possibility to shift flight earlier

' and no slot available after

' otherwhise there is other possibilities .......

lReturn = -1

Call EX\_Log(RecurLevel, " --> ERR: No slot for " & lFlightIx)

Call AU\_NPM\_MsgboxStop("ManageTimeSolution PB of NB of available slot not OK for : " & \_

lAll\_AUFlights.GetCallsignICAO(lFlightIx) & " id= " & lFlightIx, \_

lAll\_AUFlights, lMySlotsValueSorted, lMySlotsUsed, lAllFl\_nb)

End If

End If

End If

AU\_NPM\_ManageMarginPrioFlights\_ManageTimeSolution = lReturn

End Function

'----------------------------------------- AU\_NPM\_ManageMarginPrioFlights\_ManageSolutionEarlier

' Manage the solution at or before the target time or later if not found ealier

' At this stage the target slot is already used by another flight

' try to manage the solution by shifting the flight who occupy the slot earlier

' because this flight has a higher priority because it's assigned before

'Input :

' - the list of slot

' - the slot used in this list by a previous assigned flight,

' - the flight to be managed

' - the started target slot

'Output:

' - the available slot for this flight

' first check if there is an available slot before the target one (needed to find a solution earlier)

' if No available slot before : stop this function and return -1

' if there is an empty slot before (minimum constraint to have a earlier solution)

' loop from the current needed slot to the latest slot of the list

' (look also on later slot to return a slot later if no solution before)

' Call "MoveFlightEarlier" the recursive function to try to shift the flights on earlier slot

' to make a hole to assign the flight on a slot

' if a slot is found: return the slot found (end of loop)

' End of the loop (at this stage the slot found could be later)

' if no possible slot return -1

' otherwise return the slot found

Function AU\_NPM\_ManageMarginPrioFlights\_ManageSolutionEarlier( \_

lAll\_AUFlights As CL\_AllFlights, \_

ByRef lMySlotsValueSorted() As Date, ByRef lMySlotsUsed() As Integer, lAllFl\_nb As Integer, \_

lTargetSlot As Integer, \_

lMarginFlightIx As Integer) As Integer

Dim lSlotEarlierPossible As Integer

Dim lAvailableSlot\_Earlier As Integer

Dim lTargetMove As Integer

lAvailableSlot\_Earlier = -1

' to test

If lAll\_AUFlights.GetCallsignICAO(lMarginFlightIx) = "AFR165Z" Then

lMarginFlightIx = lMarginFlightIx

End If

lSlotEarlierPossible = AU\_NPMF\_GetIxOfEarlierFlightCanMove(lAll\_AUFlights, \_

lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb, lTargetSlot, lMarginFlightIx)

' some time say no but there is SO overwrite it

'lSlotEarlierPossible = 0

If lSlotEarlierPossible > -1 Then

' move flight earlier is possible

'ListRecurCall = ListRecurCall & " | T:" & lMarginFlightIx & " " & lTargetSlot & " -> "

' loop until solution found or end of slots

' apply the recursive move flight earlier function from the target slot to the end of the slots

' until a solution is found

For lTargetMove = lTargetSlot To lAllFl\_nb - 1

'ListRecurCall = "FL: " & lMarginFlightIx & " Slot: " & lTargetMove & " -> "

Call EX\_Log(RecurLevel, "Start FL: " & lMarginFlightIx & " Slot: " & lTargetMove & " -> ")

' test and manage if there is a slot earlier

lAvailableSlot\_Earlier = AU\_NPM\_ManageMarginPrioFlights\_MoveFlightEarlier(lAll\_AUFlights, \_

lMySlotsValueSorted(), lMySlotsUsed(), lAllFl\_nb, \_

lTargetMove, lMarginFlightIx)

'ListRecurCall = ListRecurCall & " Target= " & lTargetMove & " Out=" & lAvailableSlot\_Earlier")

If lAvailableSlot\_Earlier > -1 Then

' Found a place in the slot list earlierassign it

'lMySlotsUsed(lAvailableSlot\_Earlier) = lMarginFlightIx

lTargetMove = lAllFl\_nb

'ListRecurCall = ListRecurCall & " S= " & lAvailableSlot\_Earlier & " |"

Call EX\_Log(RecurLevel, "En FL: " & lMarginFlightIx & " Slot: " & lAvailableSlot\_Earlier)

Else

'ListRecurCall = ListRecurCall & " ..Next.."

Call EX\_Log(RecurLevel, " FL: " & lMarginFlightIx & " Slot Not OK .. Next")

End If

Next lTargetMove

If lAvailableSlot\_Earlier < 0 Then

' LG2018-07 No now it's OK because use of all earlier slots

' If a flight can't be on the earlier slot it's meen that ther is a slot later !!!

' SO only return -1

Call EX\_Log(RecurLevel, "En FL: " & lMarginFlightIx & " No Solution found earlier even with hole")

GoTo lNext1:

' a solution can exist

' here it's because an affected flight lock the list because it's after another one

If AU\_NPM\_ManageMarginPrioFlights\_IsFlightScheduleCompatible(lAll\_AUFlights, \_

lMySlotsValueSorted(lSlotEarlierPossible), lMarginFlightIx) = True Then

' this fligh is compatible

lAvailableSlot\_Earlier = lSlotEarlierPossible

Call EX\_Log(RecurLevel, "En FL: " & lMarginFlightIx & " Direct Slot: " & lAvailableSlot\_Earlier)

Else

' no solution because current and earlier slot not compatible with schedule

' something wrong here

Call EX\_Log(RecurLevel, "En FL: " & lMarginFlightIx & " ERROR No slot ")

Call AU\_NPM\_MsgboxStop("Margins PB possible slot before but not found solution !!! : " & \_

lAll\_AUFlights.GetCallsignICAO(lMarginFlightIx) & " id= " & lMarginFlightIx & vbCr & ListRecurCall, \_

lAll\_AUFlights, lMySlotsValueSorted, lMySlotsUsed, lAllFl\_nb)

End If

lNext1:

End If

Else

lAvailableSlot\_Earlier = -1

End If

AU\_NPM\_ManageMarginPrioFlights\_ManageSolutionEarlier = lAvailableSlot\_Earlier

End Function

' Recursive function to find and return a slot for a flight

' If the slot is not free it make it available by shifting already assign flights earlier (the already assign flights have higher priority)

' this function make the shift of the flights only if all flights can be shifted (recursive test before shifting)

'Input :

' - the list of slot

' - the slot used in this list by a previous assigned flight,

' - the flight to be managed

' - the started target slot

'Output:

' - return >-1 : the available slot position set to free for this flight

' - return = -1 blocking point, no possible shift earlier

' - return = a negative value starting at -1000

' + the negative value of the slot corresponding to a Unmovable flight

' ex: -1051 : the slot 51 is occupied by a Unmovable flight

Function AU\_NPM\_ManageMarginPrioFlights\_MoveFlightEarlier(lAll\_AUFlights As CL\_AllFlights, \_

ByRef lMySlotsValueSorted() As Date, ByRef lMySlotsUsed() As Integer, lSlots\_nb As Integer, \_

lTargetIx As Integer, lFlightIx\_ToMove As Integer) As Integer

Dim li As Integer

Dim lReturn As Integer

Dim lTo As Integer

Dim lFrom As Integer

Dim lTryPreviousFlight As Boolean

Dim lEarliestTime As Date

Dim lPos As Integer

Dim lCallsign As String

' to test

RecurLevel = RecurLevel + 1

lCallsign = lAll\_AUFlights.GetCallsignICAO(lFlightIx\_ToMove)

Call EX\_Log(RecurLevel, "Mv FL: " & lFlightIx\_ToMove & " to Slot: " & lTargetIx & "(used by " & lMySlotsUsed(lTargetIx) & ")")

' ca boucle ici !!!!!!!

If lFlightIx\_ToMove = 284 And lTargetIx = 57 And lMySlotsUsed(lTargetIx) = 272 Then

lTargetIx = lTargetIx

End If

' try to make the target slot available by shifting it earlier

lTryPreviousFlight = True

lFrom = lTargetIx

lTo = lTargetIx - 1

' initial condition

' FromSlot = Target slot

' ToSlot = Target slot - 1 slot to be use for the first shift in case the target one is used

' loop on the flights until a solution is found or no possible solution

While lTryPreviousFlight = True

' test if the flight to put on the target slot (FromSlot)is time compatible

' (with its reference time - Airport early schedule duration)

If AU\_NPM\_ManageMarginPrioFlights\_IsFlightScheduleCompatible(lAll\_AUFlights, \_

lMySlotsValueSorted(lFrom), lFlightIx\_ToMove) = False Then

'if the flight is not time compatible with the FromSLot

'stop the loop and return the index of the tested slot as negative value

' add -1000 to be sur the value 0 is well managed

lReturn = ReturnNotMovableDelta - lFrom

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, " FL: " & lFlightIx\_ToMove & " Slot: " & lFrom & " To Early .. Return : " & lReturn)

Else

'Else : the flight is compatible with the FromSlot

If lMySlotsUsed(lFrom) = -1 Then

'if the Fromslot is empty: no flight assigned on it

'return this slot and stop the loop

lReturn = lFrom

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, " FL: " & lFlightIx\_ToMove & " Slot empty OK .. Return : " & lReturn)

Else

'Else : the slot is used

'try to move the flight assigned in the used slot at an earlier position (to the ToSlot)

'if the ToSlot is < 0 (it is not possible to move before because it was the first slot)

' in this case there is no solution: and end the loop

If lTo < 0 Then

' we are at the beginning of the slot list without finding a solution

' no possible solutions return -1

lReturn = ReturnNoSolution

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, " FL: " & lFlightIx\_ToMove & " Slot -- No SOLUTION -- At the end of Slots list Return : " & lReturn)

Else

'Else: the slot is used

'try to move the flight using the ToSlot to a earlier position by using this same recursive function

'call lPos = MoveflightEarlier with in parameter: with the ToSlot target and with the flight in the FromSlot position

lPos = AU\_NPM\_ManageMarginPrioFlights\_MoveFlightEarlier(lAll\_AUFlights, \_

lMySlotsValueSorted(), lMySlotsUsed(), lSlots\_nb, \_

lTo, lMySlotsUsed(lFrom))

'if the function return a positive value : (lPos > -1)

If lPos > -1 Then

'a compatible slot has been found

'make the use of this slot effective (assign it)

'put the flight use to call the recursive function on the Slot position returned by it

'empty the slot used by it previously

'end the loop and return the empty slot

lMySlotsUsed(lPos) = lMySlotsUsed(lFrom)

lMySlotsUsed(lFrom) = -1

lReturn = lFrom 'return the slot put to free by the shift

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, " FL: " & lFlightIx\_ToMove & " Recur OK return " & lReturn & \_

" Flight move from: " & lFrom & " To: " & lPos)

ElseIf lPos = ReturnNoSolution Then

'Else if the function return a -1 value (no possible solution) : lPos = -1

' return no possible solution and close the loop

' no solution because flights impossible to move

' cannot create a hole

lReturn = ReturnNoSolution

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, " FL: " & lFlightIx\_ToMove & " Recur NO SOLUTION ")

Else

'Else : the function return a negative value < -1 a flight is blocked on its position lPos = -1xxx

'continue to loop with an earlier position

'FromSlot = Slot position used to find solution on a flight blocked :

' Slot position -1 of the flight blocked

' (be careful : corresponding to the ToSlot position when the function is called)

'ToSlot = FromSlot - 1

' test if not the earliest possible Slot to test

' If FromSlot position is > -1 , continue the loop

' otherwise stop the loop and end by a -1 solution (no solution)

lFrom = -lPos + ReturnNotMovableDelta

lTo = lFrom - 1

If lFrom < 0 Then

' From from next loop must be >= 0 otherwhise no solution, stop the loop

lTryPreviousFlight = False

lReturn = ReturnNoSolution

Call EX\_Log(RecurLevel, " FL: " & lFlightIx\_ToMove & " Recur NO solution no more Slot to check " & \_

lFrom)

Else

lTryPreviousFlight = True

Call EX\_Log(RecurLevel, " FL: " & lFlightIx\_ToMove & " Recur Next target Dde: " & \_

lTargetIx & " Check: " & lFrom)

End If

End If

End If

End If

End If

Wend

AU\_NPM\_ManageMarginPrioFlights\_MoveFlightEarlier = lReturn

RecurLevel = RecurLevel - 1

End Function

' Recursive function to find and return a slot for a flight

' If the slot is not free it make it available by shifting already assign flights earlier (the already assign flights have higher priority)

' this function make the shift of the flights only if all flights can be shifted (recursive test before shifting)

'Input :

' - the list of slot

' - the slot used in this list by a previous assigned flight,

' - the flight to be managed

' - the started target slot

'Output:

' - return >-1 : the available slot position set to free for this flight

' - return = -1 blocking point, no possible shift earlier

' - return = a negative value starting at -1000

' + the negative value of the slot corresponding to a Unmovable flight

' ex: -1051 : the slot 51 is occupied by a Unmovable flight

Function AU\_NPM\_ManageMarginPrioFlights\_MoveFlightEarlierOLD(lAll\_AUFlights As CL\_AllFlights, \_

ByRef lMySlotsValueSorted() As Date, ByRef lMySlotsUsed() As Integer, lSlots\_nb As Integer, \_

lTargetIx As Integer, lFlightIx\_ToMove As Integer) As Integer

Dim li As Integer

Dim lReturn As Integer

Dim lTo As Integer

Dim lFrom As Integer

Dim lTryPreviousFlight As Boolean

Dim lEarliestTime As Date

Dim lPos As Integer

Dim lCallsign As String

' to test

RecurLevel = RecurLevel + 1

lCallsign = lAll\_AUFlights.GetCallsignICAO(lFlightIx\_ToMove)

Call EX\_Log(RecurLevel, "Rec Start Mv FL: " & lFlightIx\_ToMove & " to Slot: " & lTargetIx & "(used by " & lMySlotsUsed(lTargetIx) & ")")

' make it available by shifting earlier previous flights

lTryPreviousFlight = True

lFrom = lTargetIx

lTo = lTargetIx - 1

' loop on the flights because some of them could not be moved earlier due to schedule

While lTryPreviousFlight = True

' test if the target slot is compatible (with reference time - Airport early schedule duration)

If AU\_NPM\_ManageMarginPrioFlights\_IsFlightScheduleCompatible(lAll\_AUFlights, \_

lMySlotsValueSorted(lFrom), lFlightIx\_ToMove) = False Then

' the flight is not compatible with the reference time of the target

' stop the loop on this flight and return the last tested flight (as negative value)

' add -1000 to be sur the value 0 is managed

' in recursive calling function, its the To target (in the current one it's the from or an earlier one)

lReturn = ReturnNotMovableDelta - lFrom

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, "Rec - FL: " & lFlightIx\_ToMove & " Slot: " & lFrom & " To Early .. Return : " & lReturn)

Else

If lMySlotsUsed(lFrom) = -1 Then

'GOOD the slot is empty

lReturn = lFrom

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, "Rec - FL: " & lFlightIx\_ToMove & " Slot empty OK .. Return : " & lReturn)

Else

' the slot is used : try to move the used slot at an earlier position

If lTo < 0 Then

' no possible solutions

' we are at the beginning of the slot list without finding a solution

lReturn = ReturnNoSolution

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, "Rec - FL: " & lFlightIx\_ToMove & " Slot -- No SOLUTION -- At the end of Slots list Return : " & lReturn)

Else

' slot used, try to move the flight using the slot to the lTo position

' use recursive call to move the used slot earlier

lPos = AU\_NPM\_ManageMarginPrioFlights\_MoveFlightEarlier(lAll\_AUFlights, \_

lMySlotsValueSorted(), lMySlotsUsed(), lSlots\_nb, \_

lTo, lMySlotsUsed(lFrom))

If lPos > -1 Then

' found a empty good slot

' make the swap slot

lMySlotsUsed(lPos) = lMySlotsUsed(lFrom)

lMySlotsUsed(lFrom) = -1

lReturn = lFrom 'return the slot put to free by the shift

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, "Rec - FL: " & lFlightIx\_ToMove & " OK return " & lReturn & \_

" Flight move from: " & lFrom & " To: " & lPos)

ElseIf lPos = ReturnNoSolution Then

' no solution because flights impossible to move

' cannot create a hole

' Normaly a later solution is possible

lReturn = ReturnNoSolution

lTryPreviousFlight = False

Call EX\_Log(RecurLevel, "Rec - FL: " & lFlightIx\_ToMove & " NO SOLUTION ")

Else

' a negative value is returned if the position is not movable

' continu to loop with an earlier position

'from is initiated and used for next test of ealier slot in this loop

lFrom = -lPos + ReturnNotMovableDelta

lTo = lFrom - 1

If lFrom < 0 Then

' From from next loop must be >= 0 therwhiqse no solution, stop the loop

lTryPreviousFlight = False

lReturn = ReturnNoSolution

Call EX\_Log(RecurLevel, "Rec - FL: " & lFlightIx\_ToMove & " NO solution no more Slot to check " & \_

lFrom)

Else

lTryPreviousFlight = True

Call EX\_Log(RecurLevel, "Rec - FL: " & lFlightIx\_ToMove & " Initial: " & lTargetIx & \_

" reloop Next target: " & lFrom)

End If

End If

End If

End If

End If

Wend

Call EX\_Log(RecurLevel, "Rec EndLevel : " & lFlightIx\_ToMove & " return : " & lReturn)

AU\_NPM\_ManageMarginPrioFlights\_MoveFlightEarlier = lReturn

RecurLevel = RecurLevel - 1

End Function

'------------------------------------------------------------ AU\_NPM\_ManageMarginPrioFlights\_AssignPrioOnlyFlight

' Assign priority only flights in the remaining slot

' list containt prio only + baseline + suspended

Sub AU\_NPM\_ManageMarginPrioFlights\_AssignOtherFlights(lAll\_AUFlights As CL\_AllFlights, \_

ByRef lSlotTime() As Date, ByRef lSlotList() As Integer, lSlot\_nb As Integer, \_

ByRef lPrioFlightSortedFl() As Integer, lPrioFlight\_nb As Integer)

'don't manage the suspended flight here

Dim lFl As Integer

Dim lFlIx As Integer

Dim lBaselineFlights() As Integer

Dim lBaselineFlightsNb As Integer

Dim lPrioOnlyFlights() As Integer

Dim lPrioOnlyFlightsNb As Integer

'Dim lSuspendedFlights() As Integer

'Dim lSuspendedFlightsNb As Integer

ReDim lBaselineFlights(lPrioFlight\_nb)

lBaselineFlightsNb = 0

ReDim lPrioOnlyFlights(lPrioFlight\_nb)

lPrioOnlyFlightsNb = 0

'ReDim lSuspendedFlights(lPrioFlight\_nb)

'lSuspendedFlightsNb = 0

For lFl = 0 To lPrioFlight\_nb - 1

lFlIx = lPrioFlightSortedFl(lFl)

If lAll\_AUFlights.GetPrio(lFlIx) = GPrioBaseline Then

lBaselineFlights(lBaselineFlightsNb) = lFlIx

lBaselineFlightsNb = lBaselineFlightsNb + 1

ElseIf lAll\_AUFlights.GetPrio(lFlIx) = GPrioSuspended Then

'lSuspendedFlights(lSuspendedFlightsNb) = lFlIx

'lSuspendedFlightsNb = lSuspendedFlightsNb + 1

Else

lPrioOnlyFlights(lPrioOnlyFlightsNb) = lFlIx

lPrioOnlyFlightsNb = lPrioOnlyFlightsNb + 1

End If

Next lFl

' ---------- manage the baseline flights on schedule

If lBaselineFlightsNb > 0 Then

Call AU\_NPM\_ManageMarginPrioFlights\_AssignBaselineFlights(lAll\_AUFlights, \_

lSlotTime(), lSlotList(), lSlot\_nb, \_

lBaselineFlights(), lBaselineFlightsNb)

End If

Erase lBaselineFlights

' ---------- Manage the prio flights

If lPrioOnlyFlightsNb > 0 Then

Call AU\_NPM\_ManageMarginPrioFlights\_AssignPrioFlights(lAll\_AUFlights, \_

lSlotTime(), lSlotList(), lSlot\_nb, \_

lPrioOnlyFlights(), lPrioOnlyFlightsNb)

End If

Erase lPrioOnlyFlights

'' ---------- Manage the Suspended flights

'If lSuspendedFlightsNb > 0 Then

' Call AU\_NPM\_ManageMarginPrioFlights\_AssignSuspendedFlights(lAll\_AUFlights, \_

' lSlotTime(), lSlotList(), lSlot\_nb, \_

' lSuspendedFlights(), lSuspendedFlightsNb)

'End If

'Erase lSuspendedFlights

End Sub

'------------------------------------------------------------ AU\_NPM\_ManageMarginPrioFlights\_AssignBaselineFlights

' Assign default Baseline flights in the remaining slot

' PB what we do if no possible slot for baseline ??????

Sub AU\_NPM\_ManageMarginPrioFlights\_AssignBaselineFlights(lAll\_AUFlights As CL\_AllFlights, \_

ByRef lSlotTime() As Date, ByRef lSlotList() As Integer, lSlot\_nb As Integer, \_

ByRef lFlightSorted() As Integer, lFlight\_nb As Integer)

Dim lFl As Integer

Dim lSlotAssigned As Integer

Dim lEarliestTime As Date

Dim lBaselineTime As Date

Dim lFlAssigned As Integer

Dim lFlIx As Integer

' ---------- manage the baseline flights on schedule

lFlAssigned = 0

' loop on baseline flights

For lFl = 0 To lFlight\_nb - 1

lFlIx = lFlightSorted(lFl)

' idem part then for Margins

' find a slot corresponding to the Margin value to put the flight

lBaselineTime = lAll\_AUFlights.GetBaselineTime(lFlIx)

lSlotAssigned = AU\_NPM\_ManageMarginPrioFlights\_ManageTimeSolution(lAll\_AUFlights, \_

lSlotTime(), lSlotList(), lSlot\_nb, \_

lBaselineTime, lFlIx)

If lSlotAssigned < 0 Then

Call AU\_NPM\_MsgboxStop("Baseline flight PB of NB of available slot not OK for : " & \_

lAll\_AUFlights.GetCallsignICAO(lFlIx) & " id= " & lFlIx, \_

lAll\_AUFlights, lSlotTime, lSlotList, lSlot\_nb)

End If

Next lFl

End Sub

'------------------------------------------------------------ AU\_NPM\_ManageMarginPrioFlights\_AssignPrioOnlyFlight

' Assign priority only flights in the remaining slot

Sub AU\_NPM\_ManageMarginPrioFlights\_AssignPrioFlights(lAll\_AUFlights As CL\_AllFlights, \_

ByRef lSlotTime() As Date, ByRef lSlotList() As Integer, lSlot\_nb As Integer, \_

ByRef lFlightSorted() As Integer, lFlight\_nb As Integer)

Dim lFl As Integer

Dim lFlIx As Integer

Dim lSlotIx As Integer

Dim lTargetTime As Date

Dim lSlotAssigned As Integer

Dim lFlightHaveSolution As Boolean

' ---------- Manage the prio flights

For lFl = 0 To lFlight\_nb - 1

lFlIx = lFlightSorted(lFl)

' Manage Prio Flights try to find a free slot compatible with the schedule

lFlightHaveSolution = False

For lSlotIx = 0 To lSlot\_nb - 1

If lSlotList(lSlotIx) = -1 Then

If AU\_NPM\_ManageMarginPrioFlights\_IsFlightScheduleCompatible(lAll\_AUFlights, \_

lSlotTime(lSlotIx), lFlIx) Then

' the schedule is compatible with the slot time

' lTime = lMySlotsValueSorted(lSlotIx)

lSlotList(lSlotIx) = lFlIx

'Call lAll\_AUFlights.SetFDATime(lFlIx, lSlotTime(lSlotIx))

lSlotIx = lSlot\_nb ' stop the loop

lFlightHaveSolution = True

End If

End If

Next lSlotIx

' test if no solution found because slot too early

If lFlightHaveSolution = False Then

' no slot available due to schedule time until the end of the slot list

lTargetTime = lAll\_AUFlights.GetHotspotEndTime

lSlotAssigned = AU\_NPM\_ManageMarginPrioFlights\_ManageTimeSolution(lAll\_AUFlights, \_

lSlotTime(), lSlotList(), lSlot\_nb, \_

lTargetTime, lFlIx)

If lSlotAssigned < 0 Then

Call AU\_NPM\_MsgboxStop("Prio flights PB of NB of available slot not OK for : " & \_

lAll\_AUFlights.GetCallsignICAO(lFlIx) & " id= " & lFlIx, \_

lAll\_AUFlights, lSlotTime, lSlotList, lSlot\_nb)

End If

End If

Next lFl

End Sub

Sub AU\_NPM\_ManageMarginPrioFlights\_AssignSuspendedFlights(lAll\_AUFlights As CL\_AllFlights, \_

ByRef lSlotTime() As Date, ByRef lSlotList() As Integer, lSlot\_nb As Integer, \_

ByRef lPrioFlightSortedFl() As Integer, lPrioFlight\_nb As Integer)

Dim lFl As Integer

Dim lFlIx As Integer

Dim ltime As Date

' get the suspended flights

For lFl = 0 To lPrioFlight\_nb - 1

lFlIx = lPrioFlightSortedFl(lFl)

If lFlIx > -1 Then ' normally never

If lAll\_AUFlights.GetPrio(lFlIx) = GPrioSuspended Then

ltime = lAll\_AUFlights.GetHotspotEndTime - G\_OneSec\_AsDate

Call lAll\_AUFlights.SetFDATime(lFlIx, ltime)

End If

End If

Next lFl

End Sub

'------------------------------------------------------------ AU\_NPM\_ManageMarginPrioFlights\_AssignPrioOnlyFlight

' Assign priority only flights in the remaining slot

Sub AU\_NPMOLD\_ManageMarginPrioFlights\_AssignSuspendedFlights(lAll\_AUFlights As CL\_AllFlights, \_

ByRef lSlotTime() As Date, ByRef lSlotList() As Integer, lSlot\_nb As Integer, \_

ByRef lFlightSorted() As Integer, lFlight\_nb As Integer)

Dim lFl As Integer

Dim lFlIx As Integer

Dim ltime As Date

' ---------- manage the baseline flights on schedule

' ---------- manage the suspended flights

For lFl = 0 To lFlight\_nb - 1

lFlIx = lFlightSorted(lFl)

If lFlIx <> -1 Then

' suspended flights at the end of the hotspot

' Dont use a slot in the middle

' the slot will be use when conpacting at the end

ltime = lAll\_AUFlights.GetHotspotEndTime - G\_OneSec\_AsDate

Call lAll\_AUFlights.SetFDATime(lFlIx, ltime)

End If

Next lFl

End Sub

'----------------------------------------------------- AU\_NPM\_ManageMarginPrioFlights\_UseAvailableSlots

' Assign priority only flights in the remaining slot

Sub AU\_NPM\_ManageMarginPrioFlights\_UseAvailableSlots(lAll\_AUFlights As CL\_AllFlights, \_

ByRef lSlotTime() As Date, ByRef lSlotList() As Integer, lSlot\_nb As Integer)

Dim lFl As Integer

Dim lFlIx As Integer

Dim lFlChg As Integer

Dim lFlChgIx As Integer

Dim lEarliestTime As Date

' compact the list

For lFl = 0 To lSlot\_nb - 1

lFlIx = lSlotList(lFl)

If lFlIx = -1 Then

' there is a hole

' find a flight to put here

For lFlChg = lFl + 1 To lSlot\_nb - 1

lFlChgIx = lSlotList(lFlChg)

If lFlChgIx > -1 Then

' there is a flight here

If AU\_NPM\_ManageMarginPrioFlights\_IsFlightScheduleCompatible(lAll\_AUFlights, \_

lSlotTime(lFl), lFlChgIx) Then

'lEarliestTime = lAll\_AUFlights.GetBaselineTime(lFlChgIx)

'If lSlotTime(lFl) >= lEarliestTime Then

' use this flight to fill the hole

'Call lAll\_AUFlights.SetFDATime(lFlChgIx, lSlotTime(lFl))

lSlotList(lFl) = lFlChgIx

lSlotList(lFlChg) = -1 ' free the slot

lFlChg = lSlot\_nb ' stop the loop

End If

End If

Next lFlChg

End If

Next lFl

End Sub