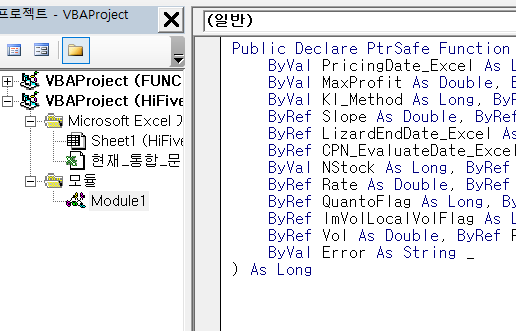
**HiFive ELS 설명서**

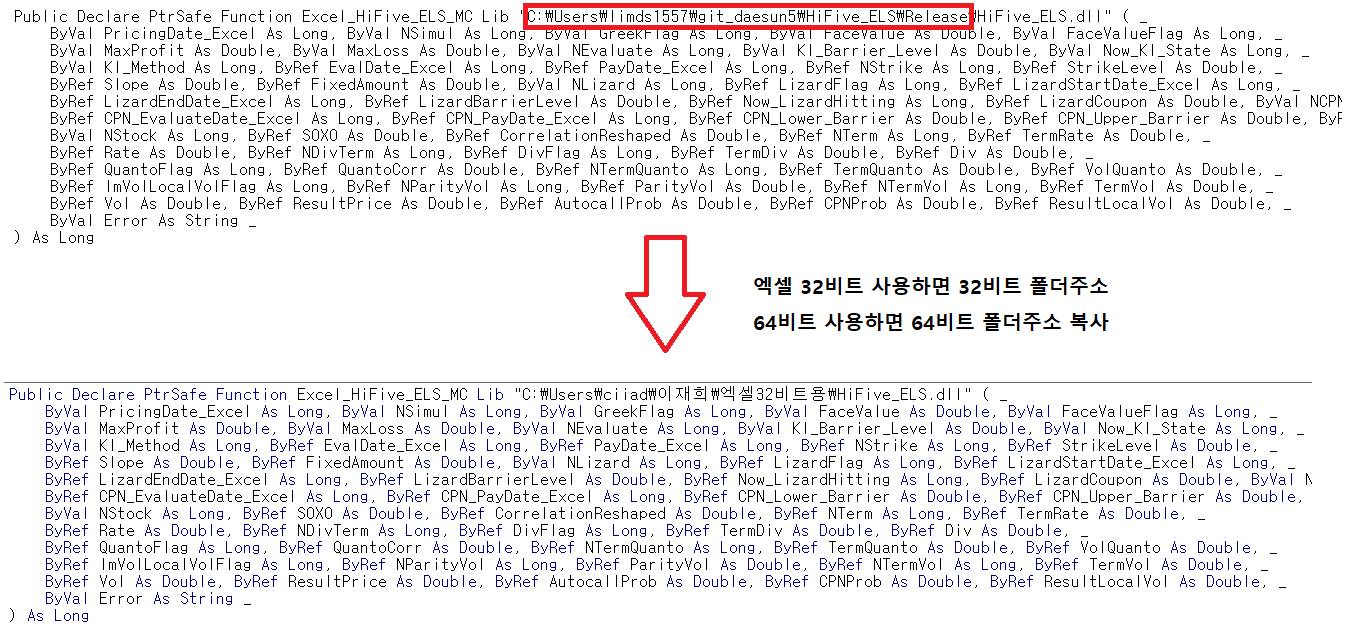
1. 자신의 엑셀 bit수 확인

파일>계정>엑셀정보

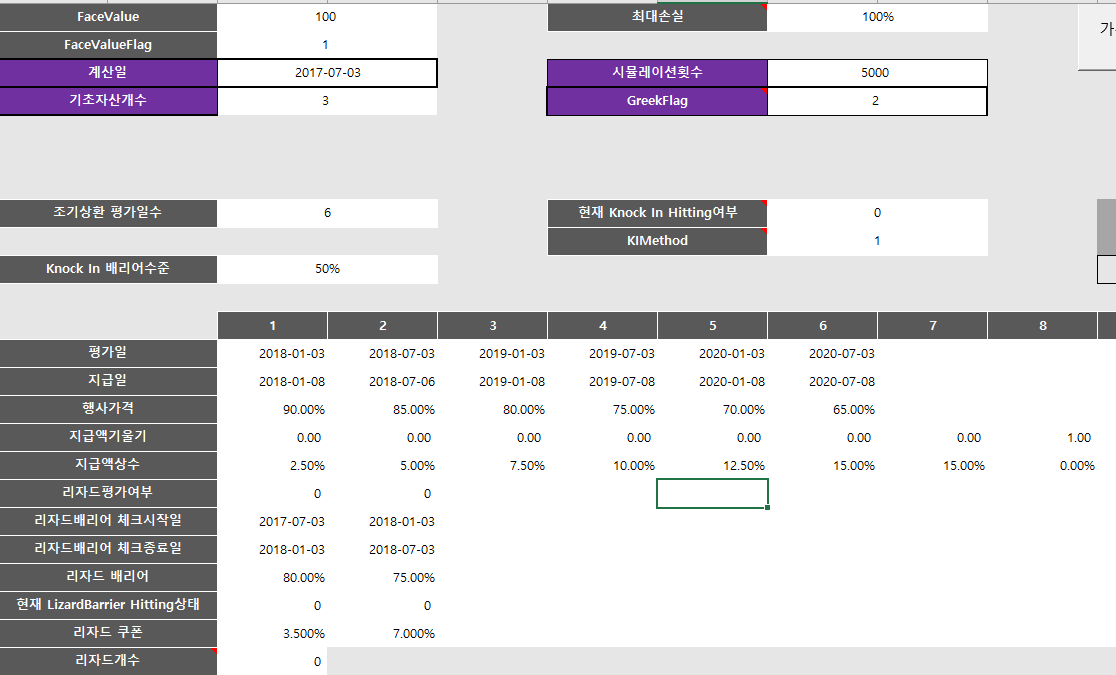


1. Alt + F11로 VBA창 키기
2. 모듈 들어가서 dll의 디렉토리 바꾸기 (현재 dll이 설치되어있는 디렉토리로 바꾸기)





1. ELS 발행정보 및 파라미터 입력하기



조기상환일 개수가 6개면 표에 6까지 입력하고

7번은 조기상환 안되었지만 낙인 이벤트 없을 경우의 Payoff

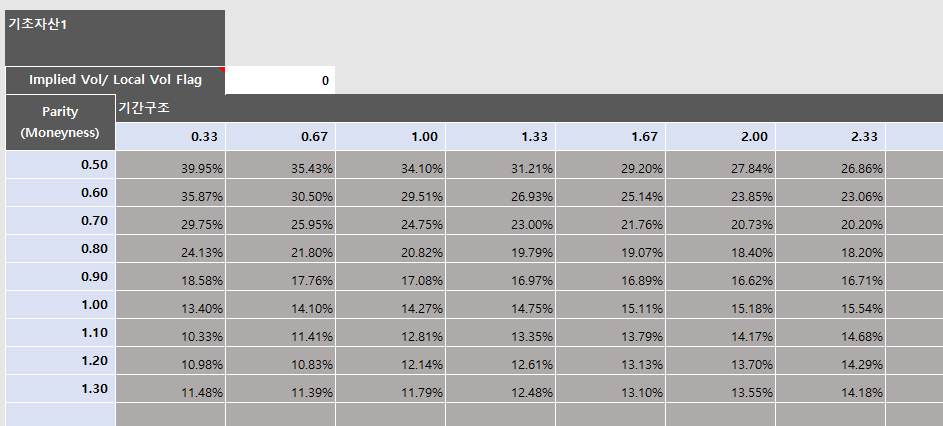
8번은 조기상환 안되었지만 낙인 이벤트 발생했을 경우 Payoff (일반적으로 주가 비례해서 손실이 나는 케이스이므로 지급액 기울기 1넣었음)

만약 리자드형 상품 아니면 리자드 개수 0넣으면 되고

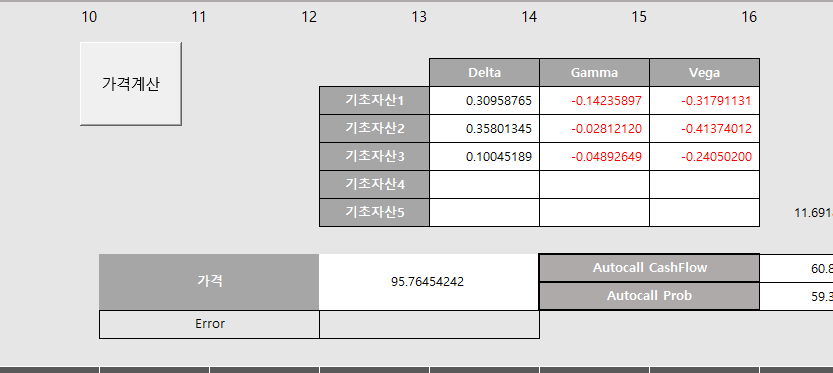
쿠폰 따로 주는 상품 아니면 쿠폰개수 0넣으면 된다.

1. 변동성 관련 입력

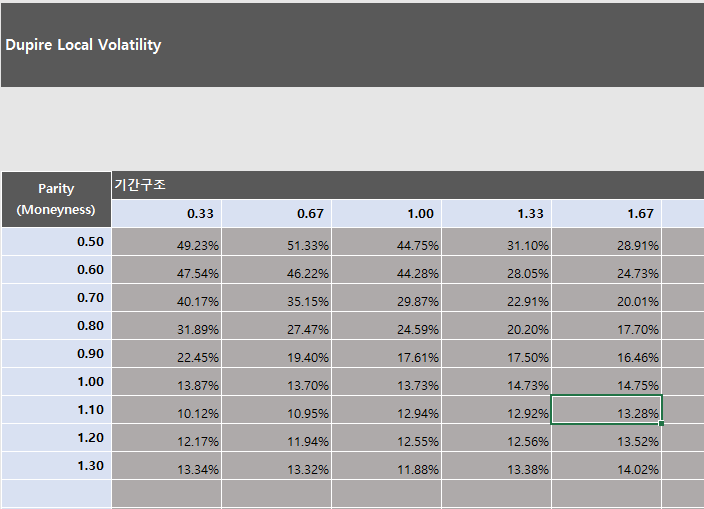
Implied Vol 블룸버그에서 뽑아서 Implied Vol Flag 0으로 넣으면 Dupire LocalVol 산출



1. 가격계산 매크로 실행



1. 가격이랑 그릭스 LocalVolatility까지 다 계산해줌



**HiFive\_ELS 함수설명서 및 클래스 매핑정의**

DLLEXPORT(long) Excel\_HiFive\_ELS\_MC(

long PricingDate\_Excel, // 가격계산날짜 ExcelType Ex) 44201(2021년01월05일)

long NSimul, // 시뮬레이션개수

long GreekFlag, // 그릭계산할지여부

double FaceValue, // 원금액

long FaceValueFlag, // 원금설정여부

double MaxProfit, // 최대이익

double MaxLoss, // 최대손실

long NEvaluate, // 조기상환 평가일개수

double KI\_Barrier\_Level, // 낙인배리어수준

long Now\_KI\_State, // 현재 낙인상태

long KI\_Method, // 낙인처리방법

long\* EvalDate\_Excel, // 조기상환 ExcelType평가일 Array [ Len =NEvaluate ]

long\* PayDate\_Excel, // 조기상환 ExcelType지급일 Array [ Len =NEvaluate ]

long\* NStrike, // 조기상환 회차별 행사가격개수 Array [ Len =NEvaluate ]

double\* StrikeLevel, // 조기상환 회차별 행사가격 Array [ Len = 3 \* NEvaluate ]

double\* Slope, // 조기상환 회차별 참여율 [ Len = 3 \* NEvaluate ]

double\* FixedAmount, // 조기상환 회차별 쿠폰 [ Len = 3 \* NEvaluate ]

long NLizard, // 리자드상환 평가일 개수

long\* LizardFlag, // 조기상환 회차별 리자드상환 평가여부 Array [Len =NLizard ]

long\* LizardStartDate\_Excel, // 조기상환회차별 리자드배리어 체크시작Array [Len=NLizard]

long\* LizardEndDate\_Excel, // 조기상환회차별 리자드배리어 체크종료일 Array [Len=NLizard]

double\* LizardBarrierLevel, // 조기상환 회차별 리자드배리어 배리어레벨 Array

long\* Now\_LizardHitting, // 조기상환 회차별 리자드배리어 현재 Hitting여부 Array

double\* LizardCoupon, // 조기상환 회차별 리자드배리어 쿠폰이자율 Array

long NCPN, // 일반 쿠폰 지급 개수

long\* CPN\_EvaluateDate\_Excel, // 일반 쿠폰 평가일 ExcelType Array [ Len = NCPN ]

long\* CPN\_PayDate\_Excel, // 일반 쿠폰 지급일 ExcelType Array [ Len = NCPN ]

double\* CPN\_Lower\_Barrier, // 일반 쿠폰 하방배리어 Array [ Len =NCPN ]

double\* CPN\_Upper\_Barrier, // 일반 쿠폰 하방배리어 Array [ Len =NCPN ]

double\* CPN\_Rate, // 일반 쿠폰 쿠폰이자율 Array [ Len =NCPN ]

long NStock, // 기초자산개수

double\* S0X0, // [자산가격,기준가격,평가일가격,Coupon기준 평가가격Array]

double\* CorrelationReshaped, // Correlation Matrix.reshape(-1)

long\* NTerm, // 이자율 Term 개수 (Disc, Rf1, Rf2, ...) [ Len= NStock +1]

double\* TermRate, // Rate Term: append(DiscT, RfT1, RfT2 + ...),Len =sum(NTerm)

double\* Rate, // Rate: append(DiscR,RfR1, RfR2, ..., Len= sum(NTerm)

long\* NDivTerm, // Div Term 개수 (S1, S2, ....) [ Len = NStock ]

long\* DivFlag, // 배당타입 (Stock1DivType, ) , Len = sum(NDivTerm)

double\* TermDiv, // 배당Term: append(S1DivTerm, S2DivTerm,..),Len= sum(NDivTerm)

double\* Div, // 배당Rate또는 금액:append(S1Div,S2Div,..),Len = sum(NDivTerm)

long\* QuantoFlag, // Quanto 사용여부 Array [Len =NStock ]

double\* QuantoCorr, // FX, 기초자산 Correlation Array [Len =NStock ]

long\* NTermQuanto, // Quanto FX Vol Term 길이 Array [ Len =NStock]

double\* TermQuanto, // FX Vol Term Array: append(FXVolT1,..) Len =sum(NTermQuanto)

double\* VolQuanto, // FX Vol Array: append[FXVol1,...] Len =sum(NTermQuanto)

long\* ImVolLocalVolFlag, // Imvol/Localvol Flag Array (0:Imvol, 1:LocalVol) Len =NStock

long\* NParityVol, // 기초자산별 패러티 개수 Array [ Shape = (NStock, )]

double\* ParityVol, // 기초자산별 패러티 Array: append[Par1 Array, Par2 Array,...]

long\* NTermVol, // 기초자산별 Volatility Term 개수 Array [ Len = NStock]

double\* TermVol, // 기초자산별 Volatility Term Array: append[VolT1, VolT2, ..]

double\* Vol, // Vol Array: append[ReshapeVol1, ReshapeVol2,…]

double\* ResultPrice, // 결과가격 및 델타감마베가 Len = (1 + NStock \* 3 )

double\* AutocallProb, // 조기상환 확률 및 조기상환 Payoff -> Len =NEvaluate \* 2

double\* CPNProb, // 쿠폰 확률 및 쿠폰 -> Len =NCPN \* 2

double\* ResultLocalVol, // LocalVolatility 결과값

)

class HiFiveInfo {

public:

double FaceValue; // Notional Amount

long FaceValueFlag; // Notional Amount 지급여부

long NStock; // Underlying Asset 개수

double MaxProfit; // 최대이익

double MaxLoss; // 최대손실

// KI 관련 정보

long KI\_Method; // KI Method 0: 낙인적용X, 1: Continuous KI, 2: 만기에만 KI

double KI\_Barrier\_Level; // Kncok In Barrier

long Now\_KI\_State; // 현재 낙인 상태

// Autocall 관련 정보

long NEvaluation; // 조기상환 평가일 개수

long\* Days\_Autocall\_Eval; // 조기상환 평가일까지 날짜수 Shape = (NEvaluation, )

long\* Days\_Autocall\_Pay; // 조기상환 지급일까지 날짜수 Shape = (NEvaluation, )

long nK; // 각 조기상환 평가일에 행사가격 개수(옵션개수)

double\*\* Strike; // 각 조기상환 평가일에 행사가격 Shape = (3, NEvaluation )

double\*\* Slope; // 각 조기상환 평가일에 참여율 Shape = (3, NEvaluation)

double\*\* FixedAmount; // 각 조기상환 평가일에 Coupon Rate Shape = (3, NEvaluation)

// Lizard 관련 정보

long NLizard; // 리자드 평가 개수

long\* LizardFlag; // 각 조기상환 평가일 리자드 평가되는지여부 Shape=(NLizard,)

long\* Days\_LizardStart; // 각 조기상환 평가일 리자드배리어평가시작일Shape=(NLizard,)

long\* Days\_LizardEnd; // 각 조기상환 평가일 리자드배리어평가종료일Shape=(NLizard,)

double\* Lizard\_Barrier\_Level; // 리자드배리어 수준 Array -> Shape = (NLizard, )

double\* Lizard\_Coupon; // 리자드 쿠폰 Array -> Shape = (NLizard, )

long\* Now\_Lizard\_KI\_State; // 현재 리자드 배리어 낙인상황 Array -> Shape = (NLizard, )

long NCPN; // 일반쿠폰평가개수

long\* Days\_CPN\_Eval; // 일반쿠폰평가일까지 날짜수 Array -> Shape = (NCPN , )

long\* Days\_CPN\_Pay; // 일반쿠폰지급일까지 날짜수 Array -> Shape = (NCPN , )

double\* CPN\_Lower\_Barrier; // 일반쿠폰지급의 하방배리어 Array -> Shape = (NCPN , )

double\* CPN\_Upper\_Barrier; // 일반쿠폰지급의 상방배리어 Array -> Shape = (NCPN , )

double\* CPN\_Rate; // 일반쿠폰Rate Array -> Shape = (NCPN , )

};

**로직설명**

1. 주가 Path 생성
2. 조기상환 및 리자드 상환 체크
3. 페이오프 할인