

## HiFive ELS 설명서

이 모델은 ELS의 가치를 계산하는 모델입니다.

DLL 경로를 찾을 수 없는 오류 발생할 경우 1~3을 따라 세팅합니다.

1. 자신의 엑셀 bit수 확인

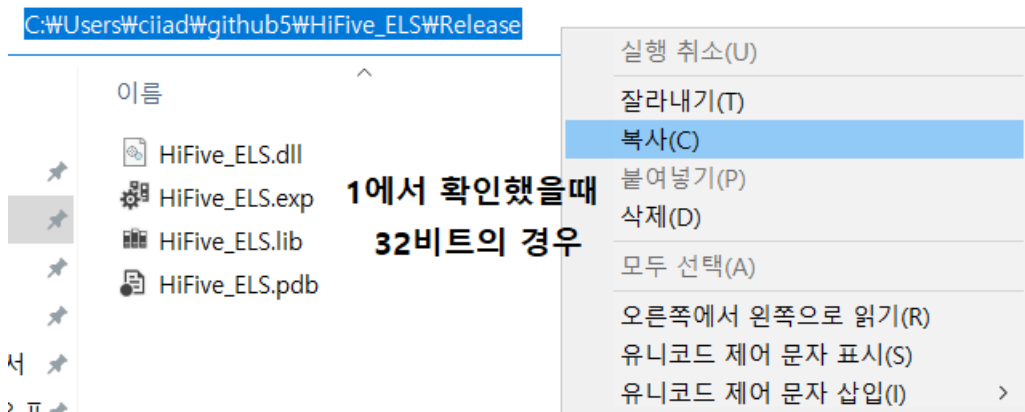
파일>계정>엑셀정보

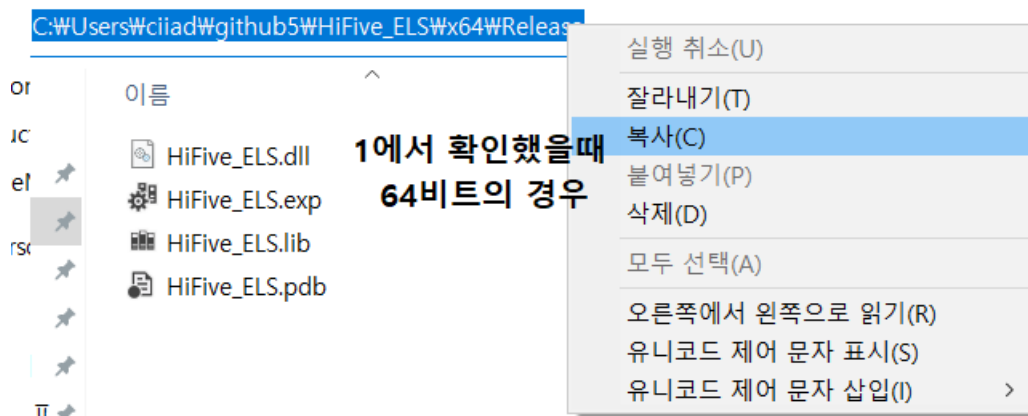


2. Alt + F11로 VBA창 키기
3. VBA 모듈 들어가서 다음 순서에 따라 dll의 디렉토리 바꾸기 (현재 dll이 설치되어있는 디렉토리로 바꾸기)

또한 엑셀이 64bit일 경우 Declare Function -> Declare PtrSafe Function으로 선언해야함

- 3-1. 엑셀 비트에 따라 경로 복사





### 3-2. 빨간 부분에 경로 붙여넣기

```
Public Declare PtrSafe Function Excel_HiFive_ELS_MC Lib "C:\Users\computer\HiFive_ELS\Release\HiFive_ELS.dll" ( _
    ByVal PricingDate_Excel As Long, ByVal NSimul As Long, ByVal GreekFlag As Long, ByVal FaceValue As Double, ByVal FaceV
    ByVal MaxProfit As Double, ByVal MaxLoss As Double, ByVal NEvaluate As Long, ByVal KL_Barrier_Level As Double, By
    ByVal KL_Method As Long, ByVal EvalDate_Excel As Long, ByVal PayDate_Excel As Long, ByVal NStrike As Long, By
    ByVal Slope As Double, ByVal FixedAmount As Double, ByVal NLizard As Long, ByVal LizardFlag As Long, ByVal L
    ByVal LizardEndDate_Excel As Long, ByVal LizardBarrierLevel As Double, ByVal Now_LizardHitting As Long, ByRef
    ByRef CPN_EvaluateDate_Excel As Long, ByRef CPN_PayDate_Excel As Long, ByRef CPN_Lower_Barrier As Double, By
    ByVal NStock As Long, ByRef SOXO As Double, ByRef CorrelationReshaped As Double, ByRef NTerm As Long, ByRef
    ByRef Rate As Double, ByRef NDivTerm As Long, ByRef DivFlag As Long, ByRef TermDiv As Double, ByRef Div As D
    ByRef QuantoFlag As Long, ByRef QuantoCorr As Double, ByRef NTermQuanto As Long, ByRef TermQuanto As Double,
    ByRef ImVolLocalVolFlag As Long, ByRef NParityVol As Long, ByRef ParityVol As Double, ByRef NTermVol As Long,
    ByRef Vol As Double, ByRef ResultPrice As Double, ByRef AutocalIProb As Double, ByRef CPNProb As Double, ByR
    ByVal Error As String _
) As Long
```



32bit의 경우 HiFive\_ELSRelease

```
Public Declare PtrSafe Function Excel_HiFive_ELS_MC Lib "C:\Users\ciiad\github5\HiFive_ELS\64\Release\HiFive_ELS.dll" ( _
    ByVal PricingDate_Excel As Long, ByVal NSimul As Long, ByVal GreekFlag As Long, ByVal FaceValue As Double, ByVal FaceV
    ByVal MaxProfit As Double, ByVal MaxLoss As Double, ByVal NEvaluate As Long, ByVal KL_Barrier_Level As Double, By
    ByVal KL_Method As Long, ByVal EvalDate_Excel As Long, ByVal PayDate_Excel As Long, ByVal NStrike As Long, By
    ByVal Slope As Double, ByVal FixedAmount As Double, ByVal NLizard As Long, ByVal LizardFlag As Long, ByVal LizardStartI
    ByRef LizardEndDate_Excel As Long, ByRef LizardBarrierLevel As Double, ByRef Now_LizardHitting As Long, ByRef LizardCoi
    ByRef CPN_EvaluateDate_Excel As Long, ByRef CPN_PayDate_Excel As Long, ByRef CPN_Lower_Barrier As Double, ByRef CPN_Upi
    ByVal NStock As Long, ByRef SOXO As Double, ByRef CorrelationReshaped As Double, ByRef NTerm As Long, ByRef TermRate A
    ByRef Rate As Double, ByRef NDivTerm As Long, ByRef DivFlag As Long, ByRef TermDiv As Double, ByRef Div As Double, _
    ByRef QuantoFlag As Long, ByRef QuantoCorr As Double, ByRef NTermQuanto As Long, ByRef TermQuanto As Double, ByRef Vol
    ByRef ImVolLocalVolFlag As Long, ByRef NParityVol As Long, ByRef ParityVol As Double, ByRef NTermVol As Long, ByRef Te
    ByRef Vol As Double, ByRef ResultPrice As Double, ByRef AutocalIProb As Double, ByRef CPNProb As Double, ByRef ResultL
    ByVal Error As String _
) As Long
```

### 4. ELS 발행정보 및 파라미터 입력하기

FaceValue	100	최대손실	100%					
FaceValueFlag	1							
계산일	2017-07-03	시뮬레이션횟수	5000					
기초자산개수	3	GreekFlag	2					
조기상환 평가일수	6	현재 Knock In Hitting여부	0					
		KIMethod	1					
Knock In 배리어수준	50%							
	1	2	3	4	5	6	7	8
평가일	2018-01-03	2018-07-03	2019-01-03	2019-07-03	2020-01-03	2020-07-03		
지급일	2018-01-08	2018-07-06	2019-01-08	2019-07-08	2020-01-08	2020-07-08		
행사가격	90.00%	85.00%	80.00%	75.00%	70.00%	65.00%		
지급액기울기	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
지급액상수	2.50%	5.00%	7.50%	10.00%	12.50%	15.00%	15.00%	0.00%
리자드평가여부	0	0						
리자드배리어 체크시작일	2017-07-03	2018-01-03						
리자드배리어 체크종료일	2018-01-03	2018-07-03						
리자드 배리어	80.00%	75.00%						
현재 LizardBarrier Hitting상태	0	0						
리자드 쿠폰	3.500%	7.000%						
리자드개수	0							

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.

## 5. 변동성 관련 입력

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

기초자산1								
Implied Vol/ Local Vol Flag		0						
Parity (Moneyess)	기간구조							
	0.33	0.67	1.00	1.33	1.67	2.00	2.33	
0.50	39.95%	35.43%	34.10%	31.21%	29.20%	27.84%	26.86%	
0.60	35.87%	30.50%	29.51%	26.93%	25.14%	23.85%	23.06%	
0.70	29.75%	25.95%	24.75%	23.00%	21.76%	20.73%	20.20%	
0.80	24.13%	21.80%	20.82%	19.79%	19.07%	18.40%	18.20%	
0.90	18.58%	17.76%	17.08%	16.97%	16.89%	16.62%	16.71%	
1.00	13.40%	14.10%	14.27%	14.75%	15.11%	15.18%	15.54%	
1.10	10.33%	11.41%	12.81%	13.35%	13.79%	14.17%	14.68%	
1.20	10.98%	10.83%	12.14%	12.61%	13.13%	13.70%	14.29%	
1.30	11.48%	11.39%	11.79%	12.48%	13.10%	13.55%	14.18%	

## 6. 가격계산 매크로 실행

10	11	12	13	14	15	16
가격계산			Delta	Gamma	Vega	
		기초자산1	0.30958765	-0.14235897	-0.31791131	
		기초자산2	0.35801345	-0.02812120	-0.41374012	
		기초자산3	0.10045189	-0.04892649	-0.24050200	
		기초자산4				
		기초자산5				11.691
가격		95.76454242	Autocall CashFlow		60.8	
			Autocall Prob		59.3	
Error						

## 7. 가격이랑 그릭스 LocalVolatility까지 다 계산해줌

# Dupire Local Volatility

Parity (Moneyness)	기간구조					
	0.33	0.67	1.00	1.33	1.67	
0.50	49.23%	51.33%	44.75%	31.10%	28.91%	
0.60	47.54%	46.22%	44.28%	28.05%	24.73%	
0.70	40.17%	35.15%	29.87%	22.91%	20.01%	
0.80	31.89%	27.47%	24.59%	20.20%	17.70%	
0.90	22.45%	19.40%	17.61%	17.50%	16.46%	
1.00	13.87%	13.70%	13.73%	14.73%	14.75%	
1.10	10.12%	10.95%	12.94%	12.92%	13.28%	
1.20	12.17%	11.94%	12.55%	12.56%	13.52%	
1.30	13.34%	13.32%	11.88%	13.38%	14.02%	

## HiFive\_ELS 함수설명서 및 클래스 매핑정의

```

DLLEXPORT(long) Excel_HiFive_ELS_MC(
    long PricingDate,           // 가격계산날짜
    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,             // 조기상환ExcelType평가일 Array [ Shape = (NEvaluate , ) ]
    long* PayDate,              // 조기상환ExcelType지급일 Array [ Shape = (NEvaluate , ) ]
    long* NStrike,              // 조기상환 회차별 행사가격개수 Array [ Shape = (NEvaluate , ) ]
    double* StrikeLevel,        // 조기상환 회차별 행사가격 Array [ Shape = reshape(3 * NEvaluate) ]

    double* Slope,              // 조기상환 회차별 참여율 [ Shape = reshape(3 * NEvaluate) ]
    double* FixedAmount,        // 조기상환 회차별 쿠폰 [ Shape = reshape(3 * NEvaluate) ]
    long NLizard,               // 리즈드상환 평가일 개수
    long* LizardFlag,           // 조기상환 회차별 리즈드상환 평가할지 여부 Array [ Shape = (NLizard , ) ]
    long* LizardStartDate,      // 조기상환 회차별 리즈드배리어 체크 시작일 ExcelType Array [ Shape = (NLizard , ) ]

    long* LizardEndDate,        // 조기상환 회차별 리즈드배리어 체크 종료일 ExcelType Array [ Shape = (NLizard , ) ]
    double* LizardBarrierLevel, // 조기상환 회차별 리즈드배리어 배리어레벨 Array
    long* Now_LizardHitting,    // 조기상환 회차별 리즈드배리어 현재 Hitting여부 Array
    double* LizardCoupon,       // 조기상환 회차별 리즈드배리어 쿠폰이자율 Array
    long NCPN,                  // 일반 쿠폰 지급 개수

    long* CPN_EvaluateDate,     // 일반 쿠폰 평가일 ExcelType Array [ Shape = (NCPN , ) ]
    long* CPN_PayDate,          // 일반 쿠폰 지급일 ExcelType Array [ Shape = (NCPN , ) ]
    double* CPN_Lower_Barrier, // 일반 쿠폰 하방배리어 Array [ Shape = (NCPN , ) ]
    double* CPN_Upper_Barrier, // 일반 쿠폰 상방배리어 Array [ Shape = (NCPN , ) ]
    double* CPN_Rate,           // 일반 쿠폰 쿠폰이자율 Array [ Shape = (NCPN , ) ]

    long NStock,                // 기초자산개수
    double* SOXO,               // [ 기초자산Array + 기준가격Array + Autocal 기준 평가가격Array + Coupon기준
    평가가격Array ]
    double* CorrelationReshaped, // Correlation Matrix.reshape(-1)
    long* NTerm,                // 이자율 기간구조 개수 (Discount, Rf1, Rf2, ....) [ Shape = (NStock +1 , ) ]
    double* TermRate,           // 이자율 기간구조 Term -> append(Discount Term Array + RfTermArray1 + RfTermArray2 + ... ) ,
    Shape = (sum(NTerm) , )

    double* Rate,               // 이자율 기간구조 Rate append(Discount Rate Array + RfRateArray1 + RfRateArray2
    + ...], Shape = (sum(NTerm), )
    long* NDivTerm,             // 배당기간구조 개수 (Stock1, Stock2, ....) [ Shape = (NStock , ) ]
    long* DivFlag,              // 배당기간구조 배당타입 (Stock1DivType, ) , Shape = (sum(NDivTerm) , )
    double* TermDiv,            // 배당기간구조 Term -> append(Stock1DivTermArray+ Stock2DivTermArray + ....) , Shape =
    (sum(NDivTerm) , )
    double* Div,                // 배당기간구조 Rate 또는 금액 -> append(Stock1DivArray + Stock2DivArray + ....),
    Shape = (sum(NDivTerm), )

```

```

long* QuantoFlag,           // Quanto 사용여부 Array [Shape = (NStock, )]
double* QuantoCorr,         // FX, 기초자산 Correlation Array [Shape = (NStock, )]
long* NTermQuanto,          // Quanto FX Vol Term 길이 Array [ Shape = (NStock, )]
double* TermQuanto,         // Quanto FX Vol Term Array -> append[ FX Vol Term1 Array + FX Vol Term2 Array + FX Vol Term3
Array + ...] [ Shape = (sum(NTermQuanto), )]
double* VolQuanto,          // Quanto FX Vol Array -> append[FX Vol1 Array + FX Vol2 Array + FX Vol3 Array + ....] [ Shape
= (sum(NTermQuanto), )]

long* ImVolLocalVolFlag,    // ImVol을 할지 LocalVol을 할지 Flag Array (0: ImVol, 1: LocalVol) [ Shape =
(NStock, )]
long* NParityVol,           // 기초자산별 패리티 개수 Array [ Shape = (NStock, )]
double* ParityVol,          // 기초자산별 패리티 Array -> append[Parity1 Array, Parity2 Array, Parity3 Array...]
long* NTermVol,             // 기초자산별 Volatility Term 개수 Array [ Shape = (NStock, )]
double* TermVol,            // 기초자산별 Volatility Term Array -> append[VolTerm1, VolTerm2, VolTerm3 ...]

double* Vol,                // 기초자산별 Volatility Array -> append[ReshapedVol1, ReshapedVol2, ReshapedVol3,
ReshapedVol4 .....]
double* ResultPrice,        // 결과가격 및 델타감마베타, SABR Parameter Shape = (1 + NStock * 3 + sum(NTermVol)*3 )
double* AutocallPrdb,      // 조기상환 확률 및 조기상환 Payoff -> Shape = (NEvaluation * 2, )
double* CPNPrdb,           // 쿠폰 확률 및 쿠폰 -> Shape = (NCPN * 2, )
double* ResultLocalVol,    // LocalVolatility 결과값

char* Error,
double* RMSPE,
long TextDumpFlag
);

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

$$dS_n = (Rf_n - d_n - \rho(S_n, FX_n)\sigma_{S(n)}\sigma_{FX(n)}) \cdot S + \sigma_{(S_n)}\sqrt{\Delta t} \times \epsilon$$

### 2. 조기상환 및 리자드 상환 체크

$$\text{if } \min(S_1, S_2, \dots) > K \text{ Then Autocall}$$

### 3. 페이오프 할인