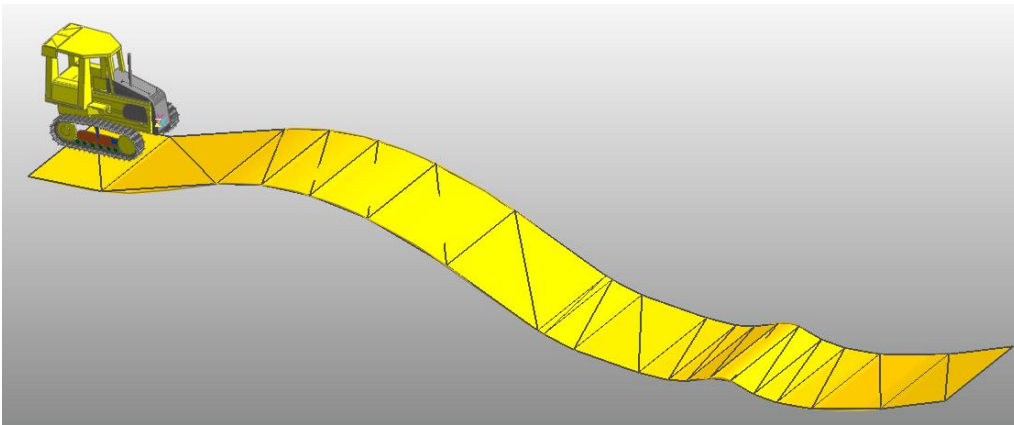




Track_LM Tutorial (eTemplate)



Copyright © 2020 FunctionBay, Inc. All rights reserved.

User and training documentation from FunctionBay, Inc. is subjected to the copyright laws of the Republic of Korea and other countries and is provided under a license agreement that restricts copying, disclosure, and use of such documentation. FunctionBay, Inc. hereby grants to the licensed user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the FunctionBay, Inc. copyright notice and any other proprietary notice provided by FunctionBay, Inc. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of FunctionBay, Inc. and no authorization is granted to make copies for such purpose.

Information described herein is furnished for general information only, is subjected to change without notice, and should not be construed as a warranty or commitment by FunctionBay, Inc. FunctionBay, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the Republic of Korea and other countries. UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

Registered Trademarks of FunctionBay, Inc. or Subsidiary

RecurDyn is a registered trademark of FunctionBay, Inc.

RecurDyn/Professional, RecurDyn/ProcessNet, RecurDyn/Acoustics, RecurDyn/AutoDesign, RecurDyn/Bearing, RecurDyn/Belt, RecurDyn/Chain, RecurDyn/CoLink, RecurDyn/Control, RecurDyn/Crank, RecurDyn/Durability, RecurDyn/EHD, RecurDyn/Engine, RecurDyn/eTemplate, RecurDyn/FFlex, RecurDyn/Gear, RecurDyn/DriveTrain, RecurDyn/HAT, RecurDyn/Linear, RecurDyn/Mesher, RecurDyn/MTT2D, RecurDyn/MTT3D, RecurDyn/Particleworks I/F, RecurDyn/Piston, RecurDyn/R2R2D, RecurDyn/RFlex, RecurDyn/RFlexGen, RecurDyn/SPI, RecurDyn/Spring, RecurDyn/TimingChain, RecurDyn/Tire, RecurDyn/Track_HM, RecurDyn/Track_LM, RecurDyn/TSG, RecurDyn/Valve are trademarks of FunctionBay, Inc.

Edition Note

This document describes the release information of **RecurDyn V9R4**.

목차


| | |
|---|----|
| 주의사항 | 5 |
| 사전 학습내용..... | 5 |
| 개 요..... | 5 |
| 과정 | 6 |
| 예상 소요 시간 | 6 |
| Template_Format Sheet 변경하기..... | 7 |
| 목적..... | 7 |
| Estimated Time to Complete | 7 |
| Template_Format Sheet 변경하기 | 8 |
| Master Sheet 생성하기 | 11 |
| 목적..... | 11 |
| 예상 소요 시간 | 11 |
| Master Sheet 생성 | 12 |
| Master Sheet 편집 | 12 |
| eTemplate HC 를 위한 Track_HC Sheet 생성하기 | 16 |
| 목적..... | 16 |
| 예상 소요 시간 | 16 |
| Track System 구성..... | 17 |
| Track_HC Sheet 완성 | 19 |
| Body 생성하기 | 24 |
| 목적 | 24 |
| 예상 소요 시간 | 24 |
| Chassis Body 생성 | 25 |
| 오른쪽 언더캐리지 생성 | 27 |
| 왼쪽 언더캐리지 생성 | 33 |
| Track Assembly 생성하기 | 41 |
| 목적 | 41 |
| 예상 소요 시간 | 41 |
| Clone Link 생성 | 42 |
| Track Assembly 생성..... | 43 |
| Joint 의 생성 및 모델 완성하기 | 47 |
| 목적..... | 47 |
| 예상 소요 시간 | 47 |
| Fixed Joint 생성..... | 48 |

| | |
|--------------------------------------|----|
| Revolute Joint 생성 | 49 |
| Ground 생성 | 53 |
| eTemplate 을 이용한 Setting 변경하기 | 55 |
| 목적 | 55 |
| 예상 소요 시간 | 55 |
| Dynamic Analysis Setting 변경 | 56 |
| Display Setting 변경..... | 56 |
| eTemplate 파일의 import 및 해석 수행하기 | 59 |
| 목적 | 59 |
| 예상 소요 시간 | 59 |
| eTemplate 파일의 Import | 60 |
| 해석 수행 | 61 |

Chapter
1

주의사항

1. 이 튜토리얼에서 Sheet 에 Excel Data 를 입력할 때, 색상으로 Header Type 과 Parameter Definition 을 구분하고 있는데, 이때 셀의 색상은 **Data** 값에는 아무런 지장을 주지 않습니다.



| | A | B | C | D |
|---|---------------------------|---------------|------------------|-------------------------|
| 1 | Header_TrackLM_Clone_Link | Name | GeometryProperty | LinkGrouserProfile |
| 2 | TrackLM_Clone_Link | LM_Clone_Link | Info_GP_Link | Info_LinkGrouserProfile |
| 3 | | | | |

2. 만약 튜토리얼 과정 중에 eTemplate 모델(엑셀파일)을 Import 했을 경우, 다시 엑셀파일을 읽어들이기 위해서는 **New Model** 을 불러온 뒤, Import 해야 합니다.
3. 튜토리얼은 각 Chapter 마다 Excel Data 를 복사해서 사용할 수 있도록 만들어져 있습니다. 왼쪽의 가위 기호 표시가 되어 있는 부분을 엑셀 Sheet 에 복사해서 사용하면 됩니다.

사전 학습내용

이 튜토리얼을 시작하기에 앞서, RecurDyn Tutorial 중 Low-mobility Tracked Vehicle Tutorial(Track_LM)을 따라 해 본 뒤, 진행하시길 바랍니다.

개요

이 튜토리얼에서는 다음과 같은 내용을 배우게 됩니다

- eTemplate 을 이용하여 Low-mobility Tracked Vehicle 을 생성
- Master Sheet 기능, HC 기능을 이해
- eTemplate 으로 Track_LM Toolkit 의 Track Body 생성 및 어셈블리

과정

튜토리얼은 다음 절차로 구성되어 있습니다. 각 절차를 완료하는 데 소요되는 예상 시간은 표에 표시됩니다.

| 과정 | 시간(분) |
|---------------------------------------|-----------|
| Template_Format Sheet 변경하기 | 5 |
| Master Sheet 생성하기 | 10 |
| eTemplate HC 를 위한 Track_HC Sheet 생성하기 | 10 |
| Body 생성하기 | 10 |
| Track Assembly 생성하기 | 10 |
| Joint 의 생성 및 모델 완성하기 | 10 |
| eTemplate 을 이용한 Setting 변경하기 | 10 |
| eTemplate 파일의 import 및 해석 수행하기 | 10 |
| Total: | 75 |



예상 소요 시간

75 분

Chapter

2

Template_Format Sheet 변경하기

목적

Template_Format Sheet 의 내용을 변경합니다.



Estimated Time to Complete

5 분

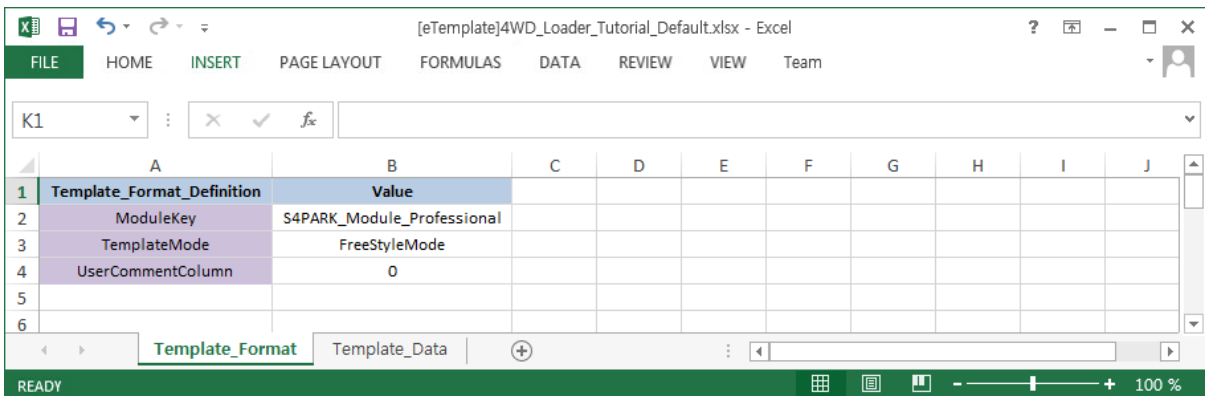
Template_Format Sheet 변경하기

Template_Format Sheet 변경:

4. 튜토리얼과 같이 배포되어 있는 **[eTemplate]Track_LM_Tutorial_Default.xlsx** 라는 이름의 eTemplate 파일을 엑셀을 이용해 엽니다.

파일 디렉토리: (<Install Dir> \Help\Tutorial\eTemplate\CreationMode\Track_LM)

5. 이 엑셀파일은 eTemplate 기본구조 파일로, **'Template_Format'**라는 이름의 Sheet 가 있습니다.

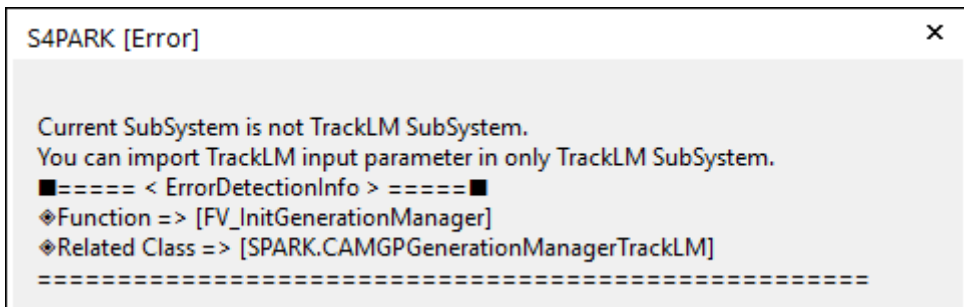
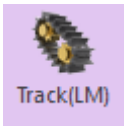


6. Track_LM Toolkit 에서 eTemplate 을 사용하기 위해 ModuleKey 설정을 변경합니다.
S4PARK_Module_Professional 을 **S4PARK_Module_TrackLM** 으로 변경하고



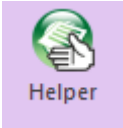
저장합니다.

7. ModuleKey 설정이 **S4PARK_Module_TrackLM** 으로 설정되어 있어야, **RecurDyn Track_LM** Toolkit 에서 eTemplate 사용이 가능하게 됩니다. 만약 ModuleKey 설정이 **S4PARK_Module_TrackLM** 으로 설정된 eTemplate 파일을 RecurDyn Track_LM



Subsystem 이 아닌 다른 Subsystem 에서 eTemplate 을 Import 하면 아래와 같이 Error Message 가 나타납니다.

Tip: eTemplate Helper 를 활용한 복사 방법



1. **Customize** 탭의 **eTemplate** 그룹에서 **Helper** 아이콘 클릭하면 **eTemplate Helper** 가 실행됩니다.

| Professional | Flexible | Track LM | Track HM |
|---------------------|------------|---------------------|---------------------|
| HC | FE Body | Track Body | Track Body |
| Body | FE Joint | Link | Link |
| Geometry | FE Force | Assembly | Assembly |
| *Geometry | FE Contact | Connector | Connector |
| Joint | FE Load | Sensor | Sensor |
| Force | | Appendix (Track LM) | Appendix (Track HM) |
| Contact | | | |
| SubEntity | | | |
| Sensor | | | |
| SubSystem | | | |
| Setting | | | |
| Analysis | | | |
| Plot | | | |
| D.O.E | | | |
| Appendix (Pro.) | | | |
| Modification (Pro.) | | | |

2. **Template Setting** 버튼을 클릭하여 Sheet 를 이동합니다.
3. **Template_Format** 의 **Header** 와 **Parameter** 를 복사합니다.
4. 각각의 **Value** 를 교재 내용에 맞게 수정합니다.

앞으로 본 교재를 진행하면서 복사해야 할 Header 와 Parameter 값도 쉽게 복사할 수 있습니다.

또는 완성본으로 제공되는 Template File 을 참조하여 쉽게 복사할 수 있습니다.

(Template File 위치:

<Install Dir> \Help\Tutorial\ETemplate\CreationMode\TrackLm\
Tracked_Vehicle_Template.xlsx)

Chapter

3

Master Sheet 생성하기

목적

Master Sheet 를 만들고, 필요한 정보를 입력합니다.



예상 소요 시간

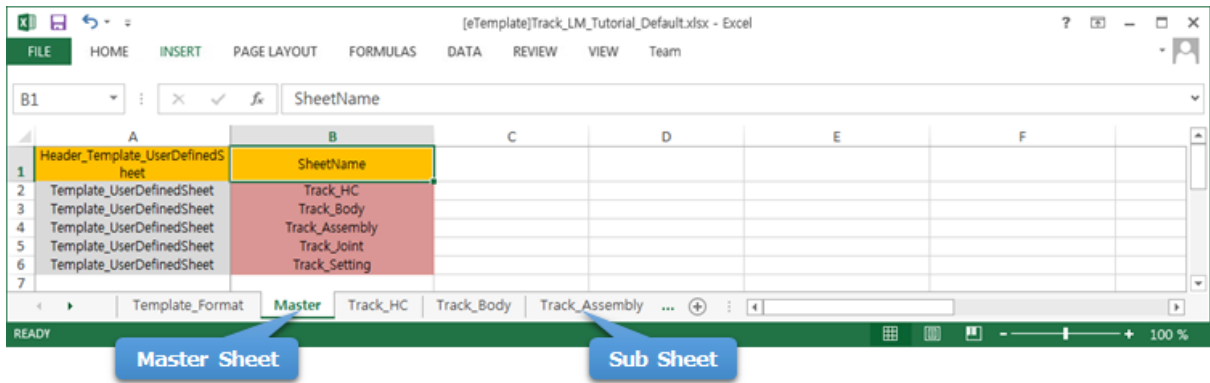
10 분

Master Sheet 생성

이 장에서는 많은 Data 로 인해 복잡한 Template_Data Sheet 를 Master Sheet 를 이용하여 구분하는 방법을 배우게 되며 생성할 Sheet 는 다음과 같습니다.

- Track_HC
- Track_Body
- Track_Assembly
- Track_Joint
- Track_Setting

Master Sheet 에는 이 엑셀파일에서 사용할 Sheet 가 무엇인지에 대한 정보를 기록할 것이며,



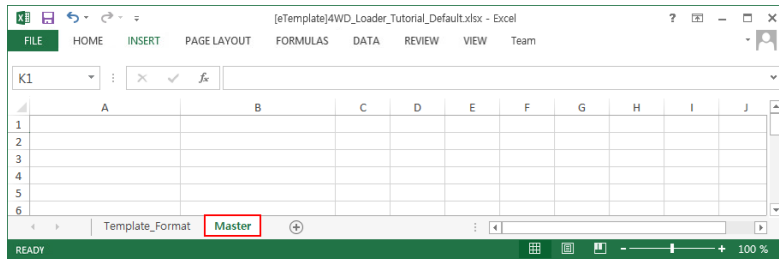
Body, Assembly, joint 정보 등을 각각 다른 Sheet 에 기록할 것입니다.

Master Sheet 편집

Template_Data Sheet 이름변경 하기:

1. **Template_Data** Sheet 의 이름을 **Master** 로 수정합니다.
2. eTemplate 기본구조에는 Master Sheet 가 있습니다. 이는 필수 Sheet 중의 하나로 사용자가 변경할 경우에 eTemplate 을 Import 할 때, 인식하지 못합니다.

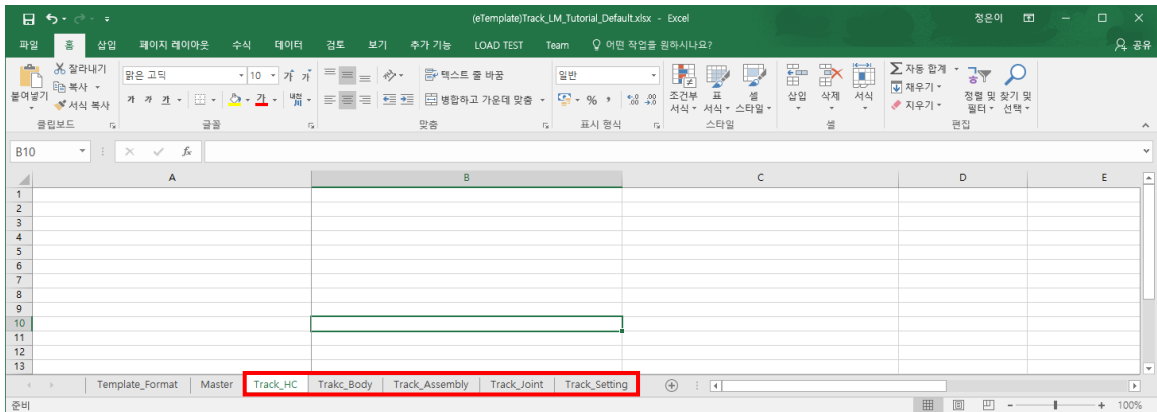
3. **Master** Sheet 는 메인 작업영역으로 설정하여 Sub Sheet 를 제어할 수 있는 기능을



추가하도록 하겠습니다.

eTemplate Sub Sheet 생성:

1. Low-mobility Tracked Vehicle 모델링을 위해 eTemplate Sub Sheet 를 생성합니다.
2. 총 5 개의 Sub Sheet 로 나누겠습니다. 이 때, 추가할 5 개의 Sheet 는 Master Sheet 를 복사해서 만들도록 합니다.
3. Master Sheet 를 5 개 복사한 후, 각각의 이름을 **Track_HC, Track_Body, Track_Assembly, Track_Joint, Track_Setting** 로 바꿉니다.



Master Sheet 구성하기:

1. Sub Sheet 를 제어하기 위해서 **Master** Sheet 에 Header 와 Parameter 를 입력합니다.
2. Header Type 에는 **Header_Template_UserDefinedSheet** 을 입력하고, 필수 파라미터는 **SheetName** 을 넣습니다.
3. Data Type 에는 **Template_UserDefinedSheet** 을 입력하고, Data 값으로 **Sub Sheet** 의 이름을 입력합니다.
4. **Master Sheet** 에 아래와 같이 입력 합니다.



5.

| Header_Template_UserDefinedSheet | SheetName |
|----------------------------------|----------------|
| Template_UserDefinedSheet | Track_HC |
| Template_UserDefinedSheet | Track_Body |
| Template_UserDefinedSheet | Track_Assembly |
| Template_UserDefinedSheet | Track_Joint |
| Template_UserDefinedSheet | Track_Setting |

위의 표를 복사하여 엑셀에 아래의 그림과 같이 붙여넣도록 합니다.

| | A | B |
|---|----------------------------------|----------------|
| 1 | Header_Template_UserDefinedSheet | SheetName |
| 2 | Template_UserDefinedSheet | Track_HC |
| 3 | Template_UserDefinedSheet | Track_Body |
| 4 | Template_UserDefinedSheet | Track_Assembly |
| 5 | Template_UserDefinedSheet | Track_Joint |
| 6 | Template_UserDefinedSheet | Track_Setting |

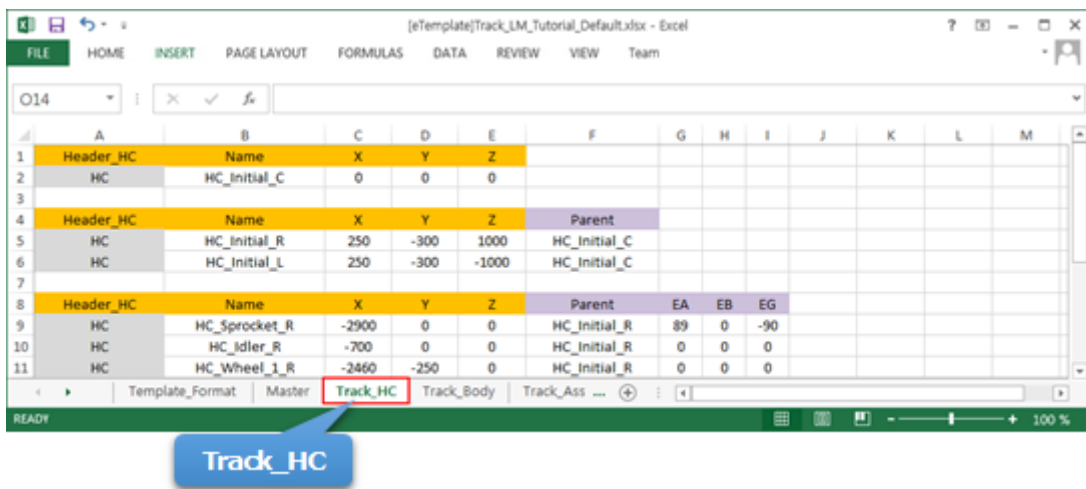
이 때, SheetName 열에 들어가는 이름은 앞서 만들었던 실제 Sheet 이름과 동일하다는 부분에 주의하도록 합니다.

Chapter
4

eTemplate HC 를 위한 Track_HC Sheet 생성하기

목적

이 장에서는 이후 장에서 사용될 eTemplate HC (Hierarchy Connector)를 생성합니다. 이 데이터들은 Track_HC Sheet 에 입력하게 됩니다.



예상 소요 시간

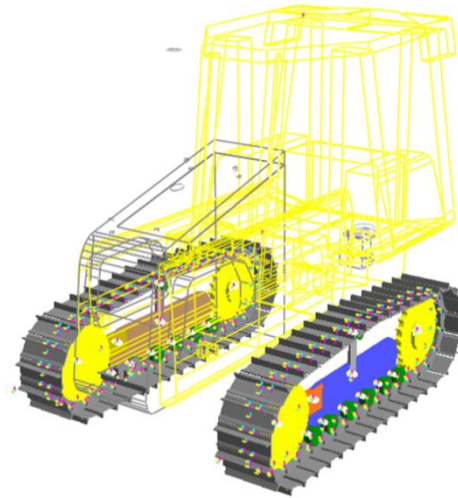
10 분

Track System 구성

Track System 구성:

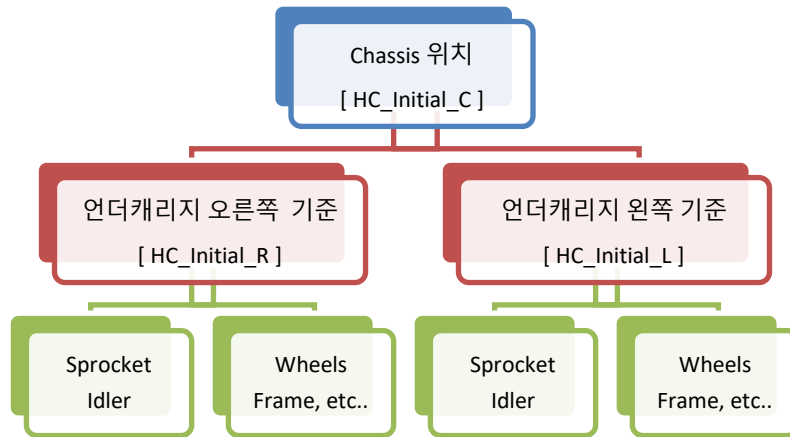
1. **Track_HC Sheet** 에 Data 를 입력하기 전에, RecurDyn Tutorial 의 Track System 을 파악해야 합니다.

(The file location : <Install Dir> \Help \Tutorial \Toolkit \Track_LM \LowMobilityTrackedVehicle)



2. **Track System** 은 크게 3 부분으로 나뉩니다. **Chassis**, **Under Carrige** (오른쪽), **Under Carrige** (왼쪽)입니다.
3. **HC (Hierarchy Connector)** 는 상대적인 위치를 결정하기 위해 사용됩니다.

4. 따라서 여기서는 Chassis의 위치를 기준으로 HC를 구성합니다. 다시 말해 Chassis의 위치가 이동하게 되면, 전체 시스템이 이동하게 됩니다. 본 튜토리얼은 아래의 그림과 같은 트리



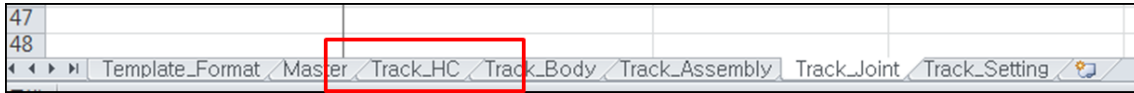
구조로 **Track System** 을 구성하였습니다.

< Track_HC Data Tree >

Track_HC Sheet 완성

Track_HC Sheet Data 입력:

1. 여기서는 Track_HC Sheet 에 입력을 합니다.



2. 가장 상위구성의 Chassis 의 위치인 HC_Initial_C 를 생성합니다.



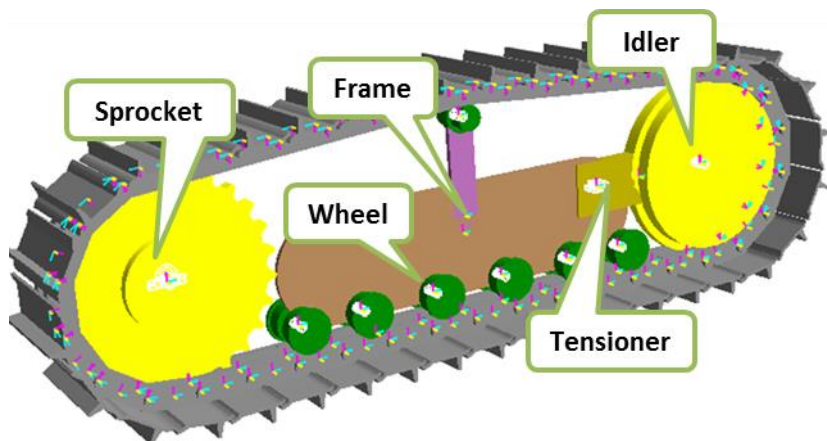
| Header_HC | Name | X | Y | Z |
|-----------|--------------|---|---|---|
| HC | HC_Initial_C | 0 | 0 | 0 |

3. Chassis 가 움직일 때, 언더캐리지의 위치도 함께 변할 수 있도록 HC 의 옵션인 Parent 를 사용하여 언더캐리지 오른쪽은 HC_Initial_R, 왼쪽은 HC_Initial_L 을 생성합니다.



| Header_HC | Name | X | Y | Z | Parent |
|-----------|--------------|-----|------|-------|--------------|
| HC | HC_Initial_R | 250 | -300 | 1000 | HC_Initial_C |
| HC | HC_Initial_L | 250 | -300 | -1000 | HC_Initial_C |

4. 오른쪽 언더캐리지의 바디들을 HC_Initial_R 을 기준으로 생성하기 위해서는 HC 옵션인



Parent 에 HC_Initial_R 를 입력합니다.

각 언더캐리지에 필요한 부품의 위치는 위의 그림과 같습니다.

(Sprocket, Idler, Wheels, Frame, Tensioner 의 자세한 위치는 RecurDyn Tutorial 참고)

5. Sprocket 의 위치(Origin)와 방향(Orientation)을 정의합니다. Sprocket 의 Teeth 와 Track Link 사이에 간섭이 일어나면 해석이 불가능합니다. 방향을 지정하여 Sprocket 을 회전시킵니다.



| Header_HC | Name | X | Y | Z | Parent | EA | EB | EG |
|-----------|---------------|-------|---|---|--------------|----|----|-----|
| HC | HC_Sprocket_R | -2900 | 0 | 0 | HC_Initial_R | 89 | 0 | -90 |

6. Sprocket 을 제외한 나머지 부품들은 위치만 고려해서 HC 를 생성합니다. 이때, Parent 로 HC_Initial_R 을 사용합니다.



| Header_HC | Name | X | Y | Z | Parent |
|-----------|-----------------------|-------|------|---|--------------|
| HC | HC_Idler_R | -700 | 0 | 0 | HC_Initial_R |
| HC | HC_Wheel_1_R | -2460 | -250 | 0 | HC_Initial_R |
| HC | HC_Wheel_2_R | -2220 | -250 | 0 | HC_Initial_R |
| HC | HC_Wheel_3_R | -1930 | -250 | 0 | HC_Initial_R |
| HC | HC_Wheel_4_R | -1640 | -250 | 0 | HC_Initial_R |
| HC | HC_Wheel_5_R | -1350 | -250 | 0 | HC_Initial_R |
| HC | HC_Wheel_6_R | -1110 | -250 | 0 | HC_Initial_R |
| HC | HC_Carrier_Roller_1_R | -1780 | 360 | 0 | HC_Initial_R |
| HC | HC_Tensioner_R | 0 | 0 | 0 | HC_Initial_R |
| HC | HC_Frame_R | 0 | 0 | 0 | HC_Initial_R |
| HC | HC_Frame_Fixed_R | -1780 | 0 | 0 | HC_Initial_R |
| HC | HC_Tensioner_Fixed_R | -1200 | 0 | 0 | HC_Initial_R |

7. 왼쪽 언더캐리지 Sprocket 의 HC 를 생성해 보겠습니다. 여기에서 **eTemplate** 의 강점이 나옵니다. 오른쪽 언더캐리지 HC 를 엑셀에서 복사하여 붙여넣기 한 뒤에 Name 과 Parent 만 HC_Initial_L 로 변경하면 됩니다.



| Header_HC | Name | X | Y | Z | Parent | EA | EB | EG |
|-----------|---------------|-------|---|---|--------------|----|----|-----|
| HC | HC_Sprocket_L | -2900 | 0 | 0 | HC_Initial_L | 89 | 0 | -90 |

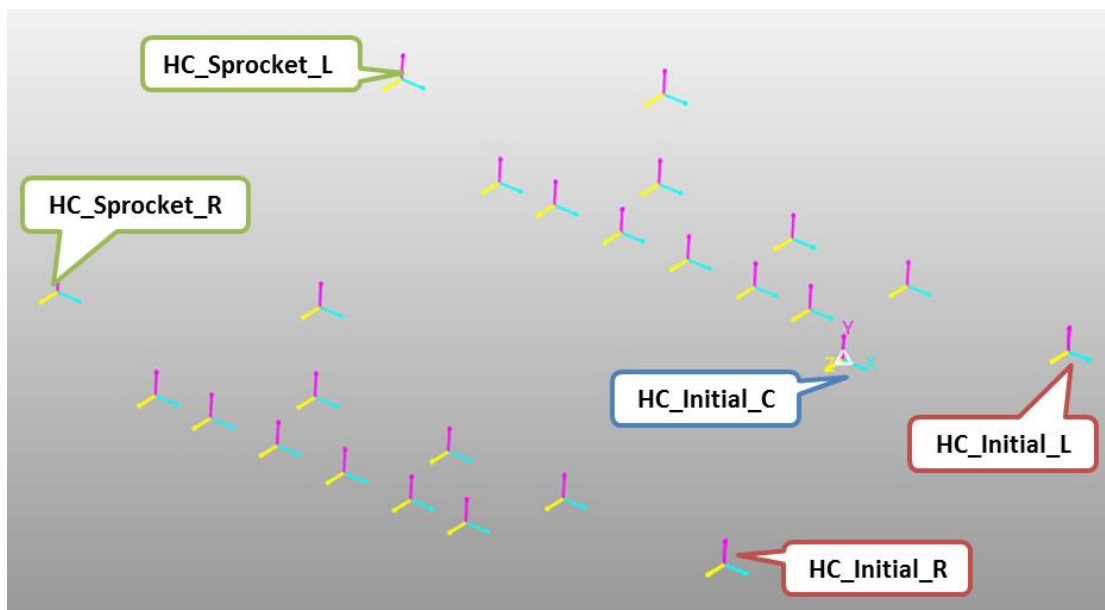
8. Sprocket 을 제외한 나머지 부품들도 복사 후, Name 과 Parent 를 변경합니다.



| Header_HC | Name | X | Y | Z | Parent |
|-----------|-----------------------|-------|------|---|--------------|
| HC | HC_Idler_L | -700 | 0 | 0 | HC_Initial_L |
| HC | HC_Wheel_1_L | -2460 | -250 | 0 | HC_Initial_L |
| HC | HC_Wheel_2_L | -2220 | -250 | 0 | HC_Initial_L |
| HC | HC_Wheel_3_L | -1930 | -250 | 0 | HC_Initial_L |
| HC | HC_Wheel_4_L | -1640 | -250 | 0 | HC_Initial_L |
| HC | HC_Wheel_5_L | -1350 | -250 | 0 | HC_Initial_L |
| HC | HC_Wheel_6_L | -1110 | -250 | 0 | HC_Initial_L |
| HC | HC_Carrier_Roller_1_L | -1780 | 360 | 0 | HC_Initial_L |
| HC | HC_Tensioner_L | 0 | 0 | 0 | HC_Initial_L |
| HC | HC_Frame_L | 0 | 0 | 0 | HC_Initial_L |
| HC | HC_Frame_Fixed_L | -1780 | 0 | 0 | HC_Initial_L |
| HC | HC_Tensioner_Fixed_L | -1200 | 0 | 0 | HC_Initial_L |

9. Track_HC Sheet 가 완성되었습니다. 이제 엑셀파일을 저장하도록 합니다.

10. 여기서 시험삼아 저장한 엑셀파일을 Import 해보도록 합니다. Import 하는 방법은



Chapter9 에 설명되어 있습니다.

Track_HC Sheet 를 eTemplate 으로 Import 하면 위와 같이 모델링 됩니다.

Chapter

5

Body 생성하기

목적

이 장에서는 다음과 같이 Body 들을 생성하게 될 것입니다.

- Chassis CAD 파일 Import
- Track Frame, Tensioner CAD 파일 Import
- Sprocket, Idler, Wheel, Carrier Roller 생성
- Track Body 들의 Geometry Information 확인



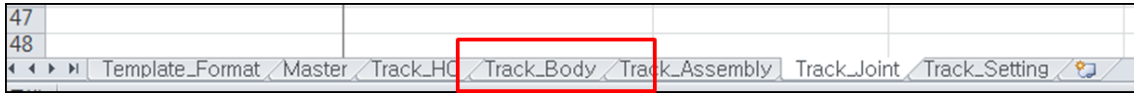
예상 소요 시간

10 분

Chassis Body 생성

Chassis CAD 파일 Import:

1. 지금부터는 **Track_Body** Sheet 에 내용을 입력합니다.

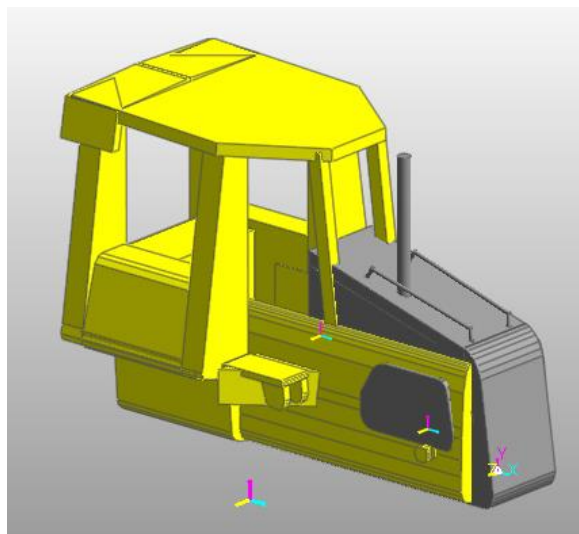


2. eTemplate 은 CAD 파일을 Import 하는 기능이 있습니다.
3. **Chassis.x_t** 를 Import 합니다.
(The file location : <Install Dir> \Help\Tutorial\eTemplate\CreationMode\Track_LM)



| Header_Body_Import | Name | RefFrame | FileName | UseAutoMerge |
|--------------------|---------|--------------|-------------|--------------|
| Body_Import | Chassis | HC_Initial_C | Chassis.x_t | True |

4. Import 의 옵션에는 2 가지가 있습니다. **UseAutoMerge** 는 CAD 파일을 Import 했을 경우에 자동적으로 하나의 Body 로 Merge 시킬 지에 대한 여부를 물어보는 옵션입니다. True 와 False 로 사용자가 옵션을 설정할 수 있습니다. 여기서는 사용되지 않지만 또 하나의 옵션은 **FilePath** 입니다. 이 옵션은 CAD 파일이 eTemplate 파일이 존재하는 폴더와 다른 곳에 있을 때, 폴더 전체 경로를 입력합니다.
5. RefFrame 은 좌표입력 방식과 참조입력 방식으로 입력이 가능합니다. 위에서 RefFrame 에 HC_Initial_C 를 입력한 것은, 앞서 Track_HC Sheet 를 이용하여 만들었던 HC_Initial_C 라는



Marker 를 Reference frame 으로 사용한다는 의미입니다.

Tip: 좌표입력 방식과 참조 입력 방식

좌표입력 방식은 Origin 과 Orientation 에 '콤마'로 구분된 숫자를 직접 입력하고, 참조입력 방식은 이미 생성한 Marker 이름을 사용하는 것입니다. (ex : HC_Initial_C)

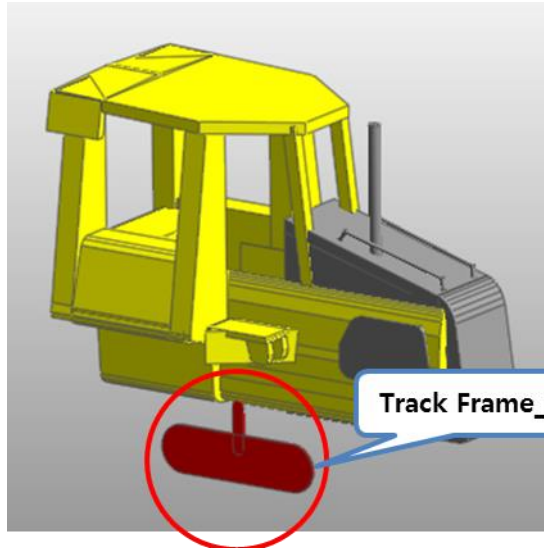
오른쪽 언더캐리지 생성

Track Frame CAD 파일 Import:

1. 언더캐리지를 구성하는 각 Track Body 가 고정될 Track Frame 을 생성합니다.
2. **Track_Frame.x_t** 를 Import 합니다.
(The file location : <Install Dir> \Help \Tutorial \Toolkit \eTemplate \CreationMode \Track_LM)
3. 오른쪽 언더캐리지의 구성으로 Track Frame 이 들어갈 수 있도록 RefFrame 에 HC_Frame_R 을 입력합니다. (Track_HC Data Tree 참고)



| Header_Body_Import | Name | RefFrame | FileName | UseAutoMerge |
|--------------------|---------------|------------|-----------------|--------------|
| Body_Import | Track_Frame_R | HC_Frame_R | Track_Frame.x_t | True |



Sprocket 생성:

1. 언더캐리지를 구성하는 Track Body 중 Sprocket 을 생성합니다.



| Header_TrackLM_Body_Sprocket | Name | RefFrame | GeometryProperty | color |
|------------------------------|------------|---------------|------------------|--------|
| TrackLM_Body_Sprocket | Sprocket_R | HC_Sprocket_R | Info_GP_Sprocket | yellow |

2. 여기서는 GeometryProperty 와 color 라는 이름의 옵션을 사용합니다. 옵션을 사용하지 않을 경우, GeometryProperty 와 color 는 기본값으로 생성됩니다.

- Sprocket 의 Geometry 파라미터를 입력하기 위해서 여기서는 GeometryProperty 라는 Information 을 사용합니다. 위에서는 다음 단계에서 만들 Info_GP_Sprocket 라는 이름의 Information 을 사용한다고 입력을 한 것입니다.
- Color 는 직접 yellow 라고 입력을 합니다.
- GeometryProperty 라는 Information 은 아래와 같이 만듭니다. 이 때, 같은 줄에 입력을 해야 합니다.



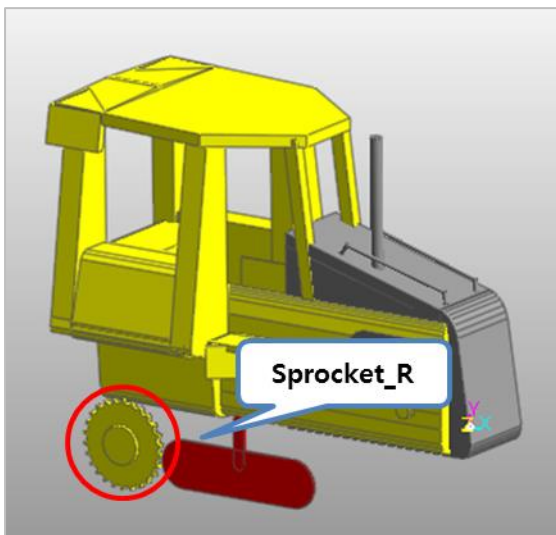
| Header_TrackLM_Info_GeometryProperty_Sprocket | Name | AddendumCircleRadius | BaseCircleRadius |
|---|------------------|----------------------|------------------|
| TrackLM_Info_GeometryProperty_Sprocket | Info_GP_Sprocket | 375 | 344 |

| DedendumCircleRadius | NumberOfTeeth | PitchCircleRadius | TrackLinkLoopRadius | TrackLinkPinCircleRadius |
|----------------------|---------------|-------------------|---------------------|--------------------------|
| 339 | 26 | 365 | 365 | 24 |

- 입력을 하면 아래와 같이 나타납니다.

| | | | | | | | | | | | | | |
|----|--|------------------|----------------------|------------------|----------------------|---------------|-------------------|---------------------|--------------------------|--|--|--|--|
| 1 | Header_Body_Import | Name | RefFrame | FileName | UseAutoMerge | | | | | | | | |
| 2 | Body_Import | Chassis | HC_Initia_C | Chassis_x_t | TRUE | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | Header_Body_Import | Name | RefFrame | FileName | UseAutoMerge | | | | | | | | |
| 5 | Body_Import | Track_Frame_R | HC_Frame_R | Track_Frame_x_t | TRUE | | | | | | | | |
| 6 | | | | | | | | | | | | | |
| 7 | Header_TrackLM_Body_Sprocket | Name | RefFrame | GeometryProperty | color | | | | | | | | |
| 8 | TrackLM_Body_Sprocket | Sprocket_R | HC_Sprocket_R | Info_GP_Sprocket | yellow | | | | | | | | |
| 9 | | | | | | | | | | | | | |
| 10 | TrackLM_Info_GeometryProperty_Sprocket | Name | AddendumCircleRadius | BaseCircleRadius | DedendumCircleRadius | NumberOfTeeth | PitchCircleRadius | TrackLinkLoopRadius | TrackLinkPinCircleRadius | | | | |
| 11 | TrackLM_Info_GeometryProperty_Sprocket | Info_GP_Sprocket | 375 | 344 | 339 | 26 | 365 | 365 | 24 | | | | |
| 12 | | | | | | | | | | | | | |

- 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어들이면 아래와 같은 모델이 생성됩니다.



Properties of Sprocket_R [Current Unit : N/kg/mm/s/deg]

| General | Graphic Property | Origin & Orientation | |
|--|------------------|--|---------------|
| Body | Contact | Geometry Data | Tooth Profile |
| Sprocket Wheel Radius(Rw) | 160. | Pv | |
| Width of Teeth(Wt) | 50. | Pv | |
| Width between Wheels(Ww) | 120. | Pv | |
| Number of Teeth | 26 | | |
| Dedendum Circle Radius(Rd) | 339. | Pv | |
| Base Circle Radius(Rb) | 344. | Pv | |
| Pitch Circle Radius(Rp) | 365. | Pv | |
| Addendum Circle Radius(Ra) | 375. | Pv | |
| Track Link Pin Circle | | | |
| Pin Circle Radius | 24. | <input type="radio"/> Assembled Radius | 267.001 |
| Loop Radius | 365. | <input checked="" type="radio"/> Radial Distance | 0. |
| <input type="radio"/> Full Search <input type="button" value="Dimension Information"/> <input type="button" value="Draw"/> <input type="button" value="Calculator"/> | | | |
| <input checked="" type="radio"/> Partial Search <input type="checkbox"/> User Boundary 0. | | | |
| <input type="button" value="Scope"/> <input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Apply"/> | | | |

Tip: eTemplate 의 Run 을 이용하여 엑셀파일을 읽어 들일 때의 주의점

Chapter 3 에서 HC (Hierarchy Connector) 를 이미 생성한 상태에서 Chapter 4 에서 저장한 엑셀 파일을 읽어 들이면 에러가 발생합니다. 그 이유는 이 엑셀파일에는 HC 의 내용도 함께 들어있기 때문에 중복해서 생성을 하려하기 때문입니다. 따라서 이번 챕터를 포함하여 각 챕터별로 제대로 입력했는지를 확인하기 위해서는 매번 새로운 Track LM 서브시스템을 만들어서 엑셀 파일을 읽어 들이도록 합니다.

Idler 생성:

1. 언더캐리지를 구성하는 Track Body 중 Idler 를 생성합니다.



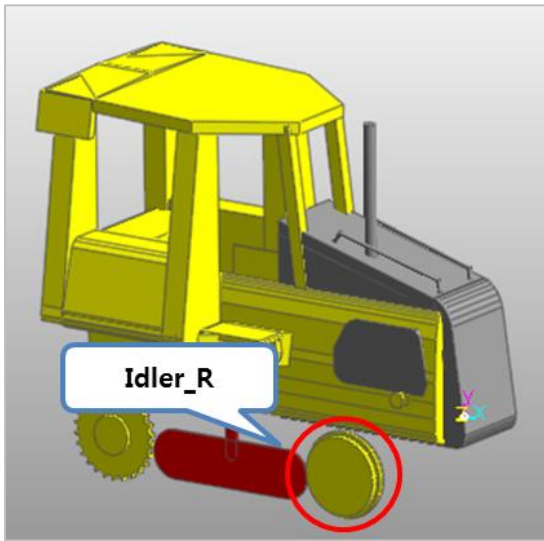
| Header_TrackLM_Body_FlangeCenter | Name | RefFrame | GeometryProperty | color |
|----------------------------------|---------|------------|------------------|--------|
| TrackLM_Body_FlangeCenter | Idler_R | HC_Idler_R | Info_GP_Idler | yellow |

2. 여기에서도 옵션으로 GeometryProperty 와 Color 를 사용합니다. 옵션을 사용하지 않으면 Idler 는 RecurDyn 의 기본값으로 생성됩니다.
3. Idler 를 위해서 Info_GP_Idler 라는 이름의 Information 을 아래와 같이 입력합니다.



| Header_TrackLM_Info_GeometryProperty_FlangeCenter | Name | InnerFlangeRadius | WheelRadius |
|---|---------------|-------------------|-------------|
| TrackLM_Info_GeometryProperty_FlangeCenter | Info_GP_Idler | 350 | 320 |

4. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Idler 과 Properties 창은 다음과 같습니다.



Properties of Idler_R [Current Unit : N/kg/mm/s/deg]

| General | Graphic Property | Origin & Orientation |
|---|--|----------------------|
| Body | Contact | Characteristics |
| Inner Flange Width(Wf) | 40. | Pv |
| Total Width(Wt) | 150. | Pv |
| Inner Flange Radius(Rf) | 350. | Pv |
| Wheel Radius(Rw) | 320. | Pv |
| Dimension Information | | |
| <input type="radio"/> Full Search | <input type="checkbox"/> User Boundary | 0. |
| <input checked="" type="radio"/> Partial Search | | |

Scope OK Cancel Apply

Wheel 생성:

1. 언더캐리지를 구성하는 Track Body 중 Wheel 을 총 6 개 생성합니다.



| Header_TrackLM_Body_FlangeSingle | Name | RefFrame | GeometryProperty | color |
|----------------------------------|---------------|--------------|------------------|-------|
| TrackLM_Body_FlangeSingle | RoadWheel_1_R | HC_Wheel_1_R | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_2_R | HC_Wheel_2_R | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_3_R | HC_Wheel_3_R | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_4_R | HC_Wheel_4_R | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_5_R | HC_Wheel_5_R | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_6_R | HC_Wheel_6_R | Info_GP_Wheel | green |

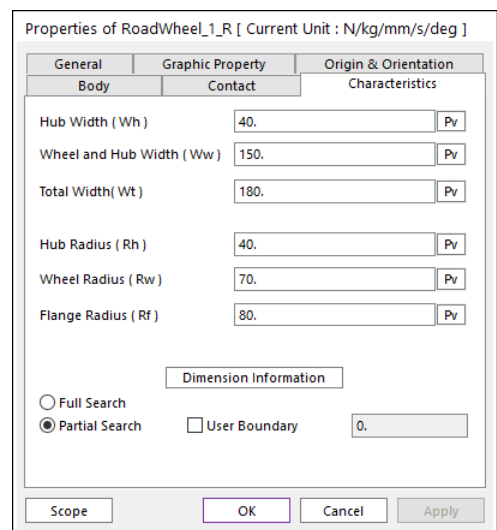
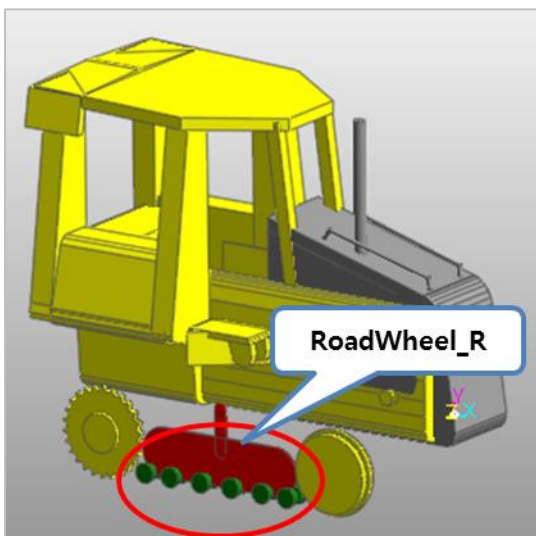
2. 여기에서도 옵션으로 GeometryProperty 와 Color 를 사용합니다. 옵션을 사용하지 않으면 Wheel 은 RecurDyn 의 기본값으로 생성됩니다.

3. Wheel 을 위해서 Info_GP_Wheel 이라는 이름의 Information 을 아래와 같이 입력합니다.



| Header_TrackLM_Info_GeometryProperty_FlangeSingle | Name | FlangeRadius | HubRadius | WheelRadius |
|---|---------------|--------------|-----------|-------------|
| TrackLM_Info_GeometryProperty_FlangeSingle | Info_GP_Wheel | 80 | 40 | 70 |

4. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Wheel 과 Properties 창은 다음과 같습니다.



Carrier Roller 생성:

1. 언더캐리지를 구성하는 Track Body 중 Carrier Roller 를 생성합니다.



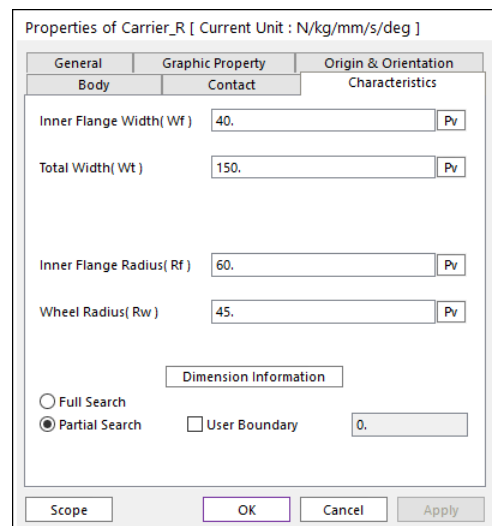
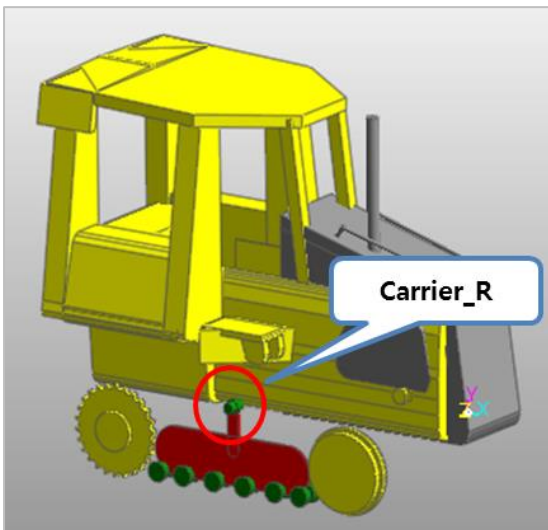
| Header_TrackLM_Body_FlangeCenter | Name | RefFrame | GeometryProperty | color |
|----------------------------------|-----------|-----------------------|------------------|-------|
| TrackLM_Body_FlangeCenter | Carrier_R | HC_Carrier_Roller_1_R | Info_GP_Carrier | green |

2. 여기에서도 옵션으로 GeometryProperty 와 Color 를 사용합니다. 옵션을 사용하지 않으면 Carrier Roller 는 RecurDyn 의 기본값으로 생성됩니다.
3. Carrier Roller 를 위해서 Info_GP_Carrier 라는 이름의 Information 을 아래와 같이 입력합니다.



| Header_TrackLM_Info_GeometryProperty_FlangeCenter | Name | InnerFlangeRadius | WheelRadius |
|---|-----------------|-------------------|-------------|
| TrackLM_Info_GeometryProperty_FlangeCenter | Info_GP_Carrier | 60 | 45 |

4. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Carrier Roller 와 Properties 창은 다음과 같습니다.



Tensioner CAD 파일 Import:

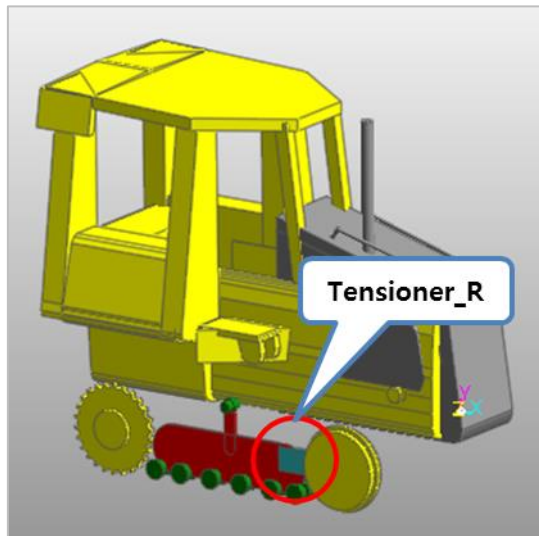
1. Idler 와 Track Frame 을 고정해 주는 Tensioner Body 를 생성합니다.
2. **Tensioner.x_t** 를 Import 합니다.

(The file location: ../Help/Tutorial/Toolkit/eTemplate/Track_LM)



| Header_Body_Import | Name | RefFrame | FileName | UseAutoMerge |
|--------------------|-------------|----------------|---------------|--------------|
| Body_Import | Tensioner_R | HC_Tensioner_R | Tensioner.x_t | True |

3. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이



생성됩니다. 이 때, 생성된 Tensioner Body 는 다음과 같습니다.

왼쪽 언더캐리지 생성

Track Frame CAD 파일 Import:

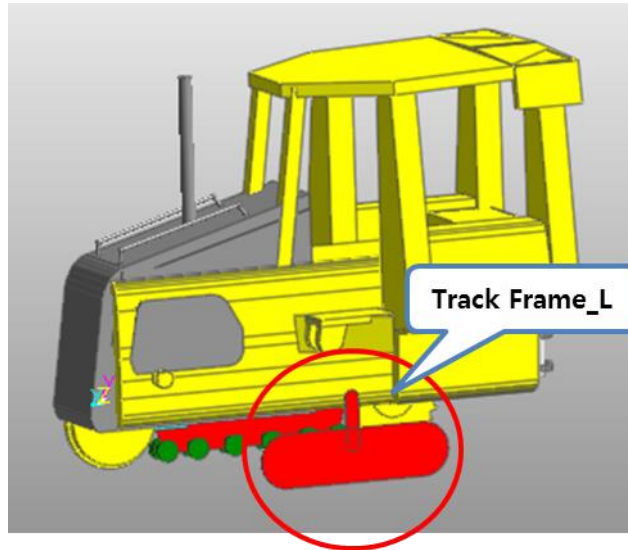
1. 언더캐리지를 구성하는 Track Body 들을 고정시킬 Track Frame 을 생성합니다.
2. **Track_Frame.x_t** 를 Import 합니다.
(The file location: ../Help/Tutorial/Toolkit/eTemplate/Track_LM)
3. Track Frame 이 오른쪽에 위치할 수 있도록 RefFrame 으로 HC_Frame_L 을 입력합니다. (Track_HC Data Tree 참고)



E T E M P L A T E L O W - M O B I L I T Y T R A C K E D V E H I C L E T U T O R I A L

| Header_Body_Import | Name | RefFrame | FileName | UseAutoMerge |
|--------------------|---------------|------------|-----------------|--------------|
| Body_Import | Track_Frame_L | HC_Frame_L | Track_Frame.x_t | True |

엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이



생성됩니다. 이 때, 생성된 Tensioner Body 는 다음과 같습니다.

Sprocket 생성:

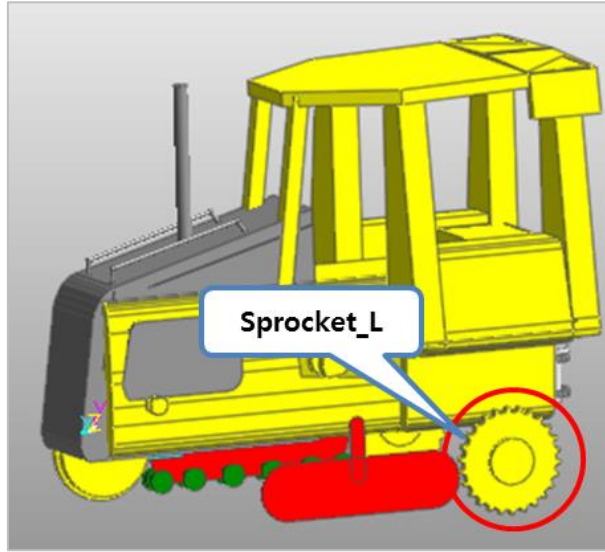
1. 언더캐리지를 구성하는 Track Body 중 Sprocket 을 생성합니다.



| Header_TrackLM_Body_Sprocket | Name | RefFrame | GeometryProperty | color |
|------------------------------|------------|---------------|------------------|--------|
| TrackLM_Body_Sprocket | Sprocket_L | HC_Sprocket_L | Info_GP_Sprocket | yellow |

2. 여기에서도 옵션으로 GeometryProperty 와 Color 를 사용합니다. 옵션을 사용하지 않으면 Sprocket 은 RecurDyn 의 기본값으로 생성됩니다.
3. 여기서 사용되는 Info_GP_Sprocket 이라는 이름의 Information 은 이미 앞단계에서 Sprocket_R 을 만들 때 입력하였으므로, 다시 입력할 필요가 없습니다.

4. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Sprocket 은 다음과 같습니다.



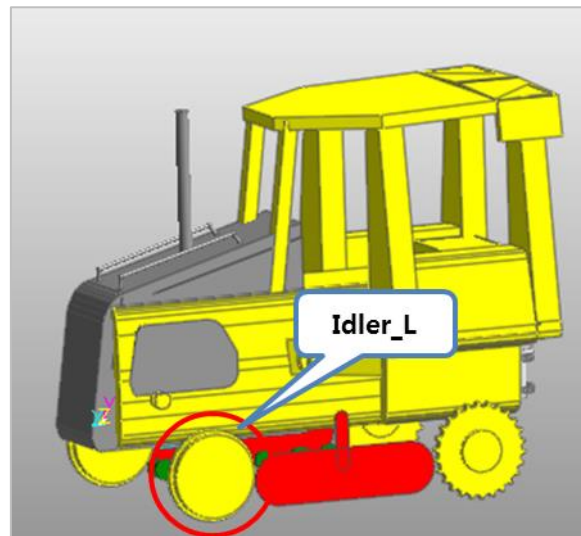
Idler 생성:

1. 언더캐리지를 구성하는 Track Body 중 Idler 를 생성합니다.



| Header_TrackLM_Body_FlangeCenter | Name | RefFrame | GeometryProperty | color |
|----------------------------------|---------|------------|------------------|--------|
| TrackLM_Body_FlangeCenter | Idler_L | HC_Idler_L | Info_GP_Idler | yellow |

2. 여기에서도 옵션으로 GeometryProperty 와 Color 를 사용합니다. 옵션을 사용하지 않으면 Idler 는 RecurDyn 의 기본값으로 생성됩니다.
3. 여기서 사용되는 Info_GP_Idler 라는 이름의 Information 은 이미 앞단계에서 Idler_R 을 만들 때 입력하였으므로, 다시 입력할 필요가 없습니다.
4. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Idler 는 다음과 같습니다.



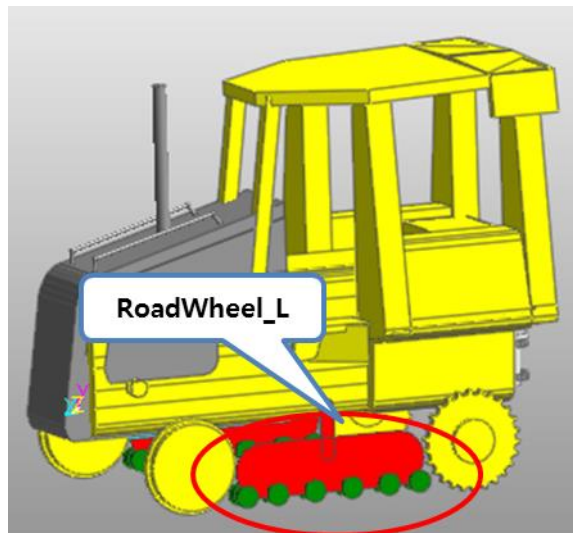
Wheel 생성:

1. 언더캐리지를 구성하는 Wheel 을 총 6 개 생성합니다.



| Header_TrackLM_Body_FlangeSingle | Name | RefFrame | GeometryProperty | color |
|----------------------------------|---------------|--------------|------------------|-------|
| TrackLM_Body_FlangeSingle | RoadWheel_1_L | HC_Wheel_1_L | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_2_L | HC_Wheel_2_L | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_3_L | HC_Wheel_3_L | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_4_L | HC_Wheel_4_L | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_5_L | HC_Wheel_5_L | Info_GP_Wheel | green |
| TrackLM_Body_FlangeSingle | RoadWheel_6_L | HC_Wheel_6_L | Info_GP_Wheel | green |

2. 옵션으로 GeometryProperty 와 Color 를 사용합니다. 옵션을 사용하지 않으면 Wheel 은 RecurDyn 의 기본값으로 생성됩니다.
3. 여기서 사용되는 Info_GP_Wheel 이라는 이름의 Information 은 이미 앞단계에서 RoadWheel_R 을 만들 때 입력하였으므로, 다시 입력할 필요가 없습니다.
4. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Wheel 은 다음과 같습니다.



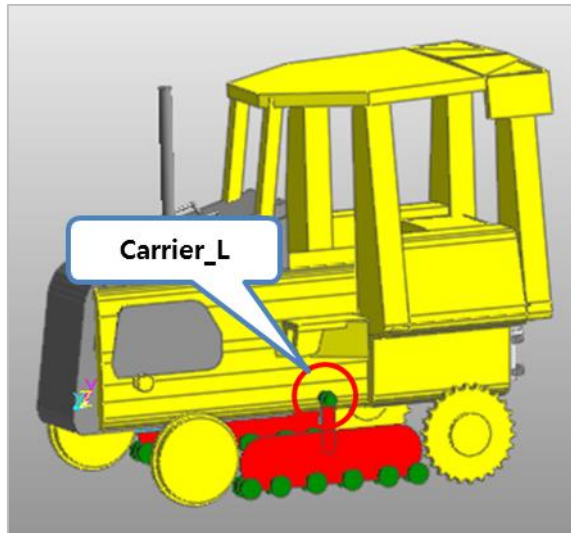
Carrier Roller 생성:

1. 언더캐리지를 구성하는 Track Body 중 Carrier Roller 를 생성합니다.



| Header_TrackLM_Body_FlangeCenter | Name | RefFrame | GeometryProperty | color |
|----------------------------------|-----------|-----------------------|------------------|-------|
| TrackLM_Body_FlangeCenter | Carrier_L | HC_Carrier_Roller_1_L | Info_GP_Carrier | green |

2. 옵션으로 GeometryProperty 와 Color 를 사용합니다. 옵션을 사용하지 않으면 Carrier Roller 는 RecurDyn 의 기본값으로 생성됩니다.
3. 여기서 사용되는 Info_GP_Carrier 라는 이름의 Information 은 이미 앞단계에서 Carrier_R 을 만들 때 입력하였으므로, 다시 입력할 필요가 없습니다.
4. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Carrier 는 다음과 같습니다.

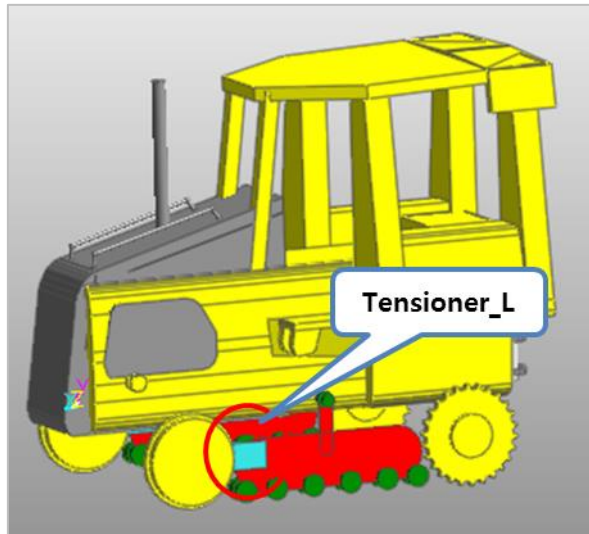


Tensioner CAD 파일 Import:

1. Idler 와 Track Frame 을 고정해 주는 Tensioner Body 를 생성합니다.
2. **Tensioner.x_t** 를 Import 합니다.
(The file location: ../Help/Tutorial/Toolkit/eTemplate/Track_LM)
3. Tensioner 가 오른쪽에 위치할 수 있도록 RefFrame 에 HC_Tensioner_R 을 입력합니다.
(Track_HC Data Tree 참고)
4. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Tensioner 는 다음과 같습니다.



| Header_Body_Import | Name | RefFrame | FileName | UseAutoMerge |
|--------------------|-------------|----------------|---------------|--------------|
| Body_Import | Tensioner_L | HC_Tensioner_L | Tensioner.x_t | True |



이로써 좌우측의 언더캐리지가 완성되었습니다.

Track Assembly 생성하기

목적

이 장에서는 Link 를 생성하고 Track Assembly 를 생성하게 될 것입니다.

- Track Link 생성
- Track Assembly 생성



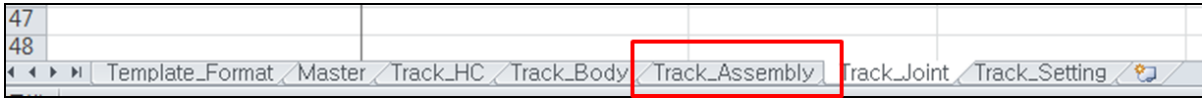
예상 소요 시간

10 분

Clone Link 생성

Clone Link 생성:

1. 지금부터는 **Track_Assembly** Sheet 에 내용을 입력합니다.



2. Track Assembly 을 하기 위해서는 Clone Link 를 먼저 만들어야 합니다.
3. 아래와 같이 입력하여 Clone Link 를 생성합니다.



| Header_TrackLM_Clone_Link | Name | GeometryProperty | LinkGrouserProfile |
|---------------------------|---------------|------------------|-------------------------|
| TrackLM_Clone_Link | LM_Clone_Link | Info_GP_Link | Info_LinkGrouserProfile |

4. 옵션으로 **GeometryProperty** 와 **LinkGrouserProfile** 를 사용합니다. 옵션을 사용하지 않으면 Clone Link 는 RecurDyn 의 기본값으로 생성됩니다.
5. 아래와 같이 **Info_GP_Link** 라는 이름의 Information 을 생성합니다. 아래는 같은 줄에 입력해야 합니다.



| Header_TrackLM_Info_GeometryProperty_Link | Name | GrouserWidth | LeftLength |
|---|--------------|--------------|------------|
| TrackLM_Info_GeometryProperty_Link | Info_GP_Link | 460 | 119.5 |

| PinRadius | RightLength | LeftPinPosition | RightPinPosition |
|-----------|-------------|-----------------|------------------|
| 24 | 119.5 | -88.5,39 | 88.5,39 |

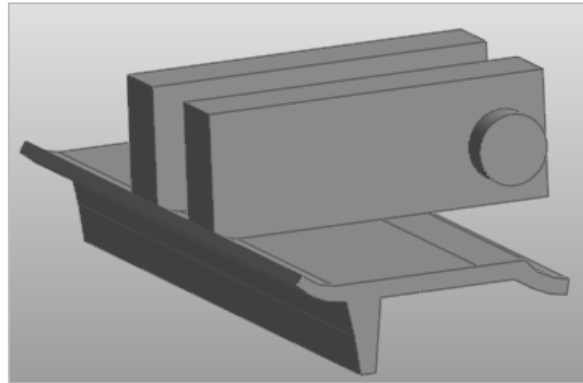
6. **FileNameForGrouserProfile** 은 Clone Link 에 사용될 Grouser 의 형상을 나타내는 프로파일입니다. 이는 Grouser1.mat 이라는 이름의 파일을 가져와서 사용하게 되며, 이 파일은 엑셀파일과 동일한 위치에 있어야 합니다.



| Header_TrackLM_Info_LinkGrouserProfile | Name | FileNameForGrouserProfile |
|--|-------------------------|---------------------------|
| TrackLM_Info_LinkGrouserProfile | Info_LinkGrouserProfile | Grouser1.mat |

7. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Link 는 다음과 같습니다. (이 때, 아래와 같은 모델을 확인하는

용도의 경우에는 앞 챕터에서 작성한 Track_Body Sheet 의 내용을 모두 지운 후 확인하십시오. 물론 전체 모델 작성을 위해서는 Track_Body Sheet 가 반드시 필요하므로, 백업을 해둔 후, 지우시기 바랍니다.)



Track Assembly 생성

오른쪽 **Track Assembly** 생성:

1. Track Assembly 생성에는 앞서 생성한 Clone Link 가 사용됩니다.
2. 필수 Parameter 인 **PassingEntityList** 에는 Track Assembly 를 만들 때 지나가는 Track Body 를 순서대로 입력합니다. 각 Body 에 대한 구분은 '/'로 표시합니다.
3. Passing Entity 는 Sprocket → RoadWheel (1~6) → Idler → Carrier 순서로 만듭니다.



| Header_TrackLM_Assembly | Name | TrackLinkClone | PassingEntityList | InOutList |
|-------------------------|---------------|----------------|--|---|
| TrackLM_Assembly | LM_Assembly_R | LM_Clone_Link | Sprocket_R/ RoadWheel_1_R/ RoadWheel_2_R/ RoadWheel_3_R/ RoadWheel_4_R/ RoadWheel_5_R/ RoadWheel_6_R/ Idler_R/ Carrier_R | Out/ Out/ Out/ Out/ Out/ Out/ Out/ Out/ Out |

4. 필수 Parameter 인 **InOutList** 는 Link 가 안쪽 방향으로 지나갈지 바깥쪽 방향을 지나갈지를 결정해 줍니다. 이때, PassingEntityList 의 Data 개수와 InOutList Data 개수가 동일해야 합니다.

5. 옵션으로 BushingForceParameter 를 입력합니다



BushingForceParameter

Info_BushingForce

지금까지 입력한 값은 엑셀파일에 아래와 같이 표시됩니다.

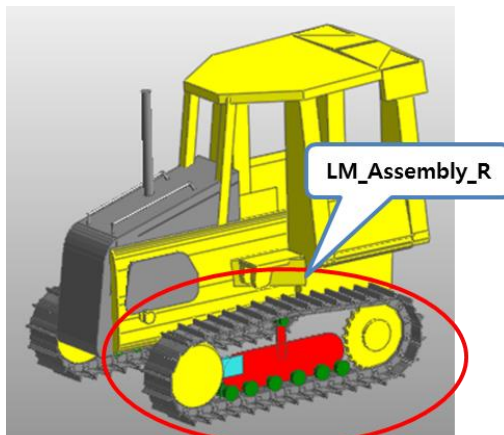
| | A | B | C | D | E | F |
|----|---|-------------------------|---------------------------|--|---------------------------------|-----------------------|
| 1 | Header_TrackLM_Clone_Link | Name | GeometryProperty | LinkGrouserProfile | | |
| 2 | TrackLM_Clone_Link | LM_Clone_Link | Info_GP_Link | Info_LinkGrouserProfile | | |
| 3 | | | | | | |
| 4 | Header_TrackLM_Info_GeometryProperty_Link | Name | GrouserWidth | LeftLength | PinRadius | RightLength |
| 5 | TrackLM_Info_GeometryProperty_Link | Info_GP_Link | 460 | 119.5 | 24 | 119.5 |
| 6 | | | | | | |
| 7 | Header_TrackLM_Info_LinkGrouserProfile_Link | Name | FileNameForGrouserProfile | | | |
| 8 | TrackLM_Info_LinkGrouserProfile_Link | Info_LinkGrouserProfile | Grouser1.mat | | | |
| 9 | | | | | | |
| 10 | Header_TrackLM_Assembly | Name | TrackLinkClone | PassingEntityList | InOutList | BushingForceParameter |
| 11 | TrackLM_Assembly | LM_Assembly_R | LM_Clone_Link | Sprocket_R/RoadWheel_1_R/RoadWheel_2_R/RoadWheel_3_R/RoadWheel_4_R/RoadWheel_5_R/RoadWheel_6_R/Idler_R/Carrier_R | Out/Out/Out/Out/Out/Out/Out/Out | Info_BushingForce |

- 아래와 같이 BushingForceParameter 로 사용될 Information 을 Info_BushingForce 라는 이름으로 생성합니다.



| Header_TrackLM_Info_Assembly_BushingForce | Name | RotationPresetAngle |
|---|-------------------|---------------------|
| TrackLM_Info_Assembly_BushingForce | Info_BushingForce | 10 |

- 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Track Assembly 는 다음과 같습니다. (만일 앞서 Clone Link 를 확인하기 위해 Track_Body Sheet 을 지웠다면, 이번 입력을 확인할 때는 다시 Track_Body Sheet 이 있는 상태에서 확인을 하시기 바랍니다. 이번에 입력한 PassingEntityList 에서는



Track_Body Sheet 에 입력된 Sprocket_R 등이 사용됩니다.)

왼쪽 **Track Assembly** 생성:

1. 필수 Parameter 인 PassingEntityList 에는 Track Assembly 를 만들 때 지나가는 Track Body 를 순서대로 입력합니다. 각 Body 에 대한 구분은 '/'로 표시합니다.
2. Passing Entity 입력시 오른쪽 언더캐리지 Track Body 이름을 사용하시면 안됩니다.



| Header_TrackLM_Assembly | Name | TrackLinkClone | PassingEntityList | InOutList |
|-------------------------|---------------|----------------|--|---|
| TrackLM_Assembly | LM_Assembly_L | LM_Clone_Link | Sprocket_L/ RoadWheel_1_L/ RoadWheel_2_L/ RoadWheel_3_L/ RoadWheel_4_L/ RoadWheel_5_L/ RoadWheel_6_L/ Idler_L/ Carrier_L | Out/ Out/ Out/ Out/ Out/ Out/ Out/ Out/ Out |

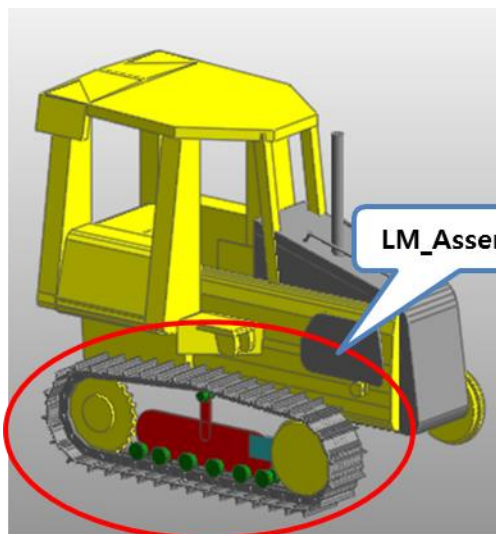
3. 앞서와 마찬가지로 PassingEntityList 의 Data 개수와 InOutList Data 개수가 동일해야 합니다.
4. 옵션으로 BushingForceParameter 를 입력합니다.



BushingForceParameter

Info_BushingForce

5. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 Assembly 는



때, 생성된 Track
다음과 같습니다.

Joint 의 생성 및 모델 완성하기

목적

이 장에서는 다음과 같이 몇 가지 Joint 들을 생성하게 될 것입니다.

- Chassis 와 Track Frame 사이의 Fixed Joint
- Track Body 와 Track Frame 사이의 Revolute Joint
- Idler 와 Tensioner 사이의 Revolute Joint
- Slider 와 Grounded Bracket 사이의 Translational Joint
- Ground 의 생성



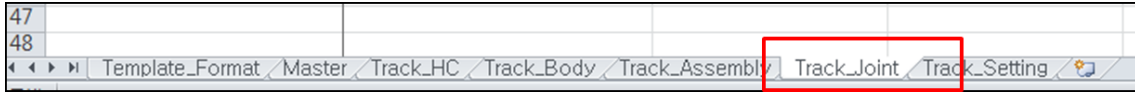
예상 소요 시간

10 분

Fixed Joint 생성

Chassis 와 **Track Frame** 사이의 **Fixed Joint** 생성:

1. 지금부터는 **Track_Joint** Sheet 에 내용을 입력합니다

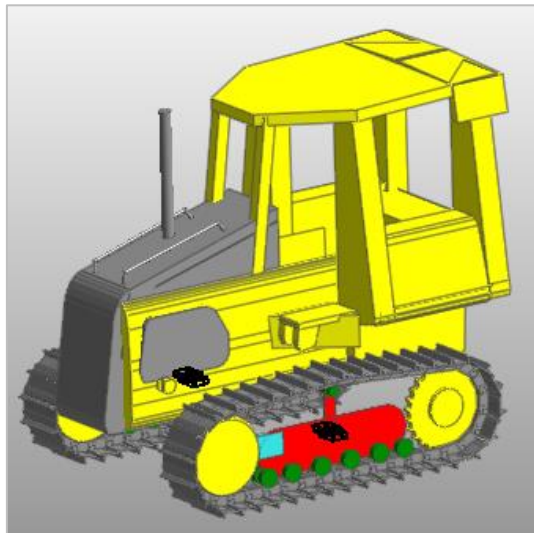


2. 여기서는 Chassis 와 양쪽 Track Frame 을 고정하기 위한 2 개의 Fixed Joint 를 생성합니다.
3. BaseEntity 로 Chassis 를, ActionEntity 로 각각 Frame Body 인 Track_Frame_R 와 Track_Frame_L 을 입력합니다. Base 와 Action 이 바뀌지 않도록 주의하시기 바랍니다.
4. RefFrame 에는 Track_HC Sheet 에 미리 입력해둔 HC_Frame_Fixed_R 과 HC_Frame_Fixed_L 을 입력합니다.



| Header_JointEx_Fixed | Name | BaseEntity | ActionEntity | RefFrame |
|----------------------|-------------------|------------|---------------|------------------|
| JointEx_Fixed | Fix_Track_Frame_R | Chassis | Track_Frame_R | HC_Frame_Fixed_R |
| JointEx_Fixed | Fix_Track_Frame_L | Chassis | Track_Frame_L | HC_Frame_Fixed_L |

5. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이



생성됩니다. 이 때, 생성된 Joint 는 다음과 같습니다.

Track Frame 와 **Tensioner** 사이의 **Fixed Joint** 생성:

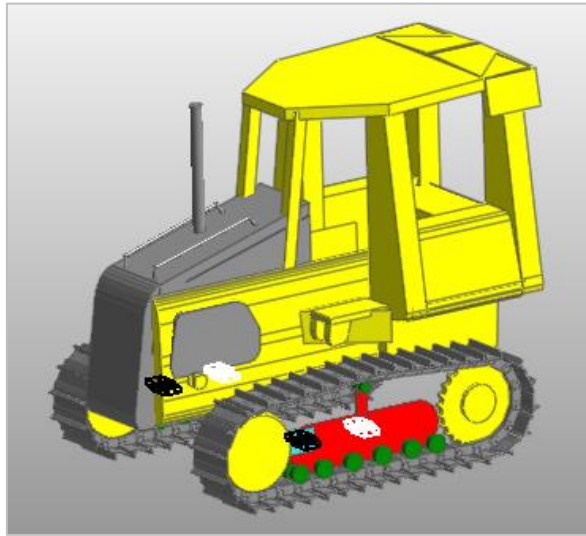
1. Track Frame 과 양쪽 Tensioner 를 고정하기 위한 2 개의 Fixed Joint 를 생성합니다.

2. BaseEntity 로 Track_Frame 이 들어가고, ActionEntity 는 Tensioner 를 입력합니다.
3. RefFrame 에는 HC_Tensioner_Fixed_R 과 HC_Tensioner_Fixed_L 을 각각 입력합니다.



| Header_JointEx_Fixed | Name | BaseEntity | ActionEntity | RefFrame |
|----------------------|-----------------|---------------|--------------|----------------------|
| JointEx_Fixed | Fix_Tensioner_R | Track_Frame_R | Tensioner_R | HC_Tensioner_Fixed_R |
| JointEx_Fixed | Fix_Tensioner_L | Track_Frame_L | Tensioner_L | HC_Tensioner_Fixed_L |

4. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이



생성됩니다. 이 때, 생성된 Joint 는 다음과 같습니다.

Revolute Joint 생성

Track Frame 과 **Sprocket** 사이의 **Revolute Joint** 생성과 모션 정의:

1. Frame Body 에 각 Spocket 을 Revolute Joint 로 고정합니다.
2. BaseEntity 로 Track_Frame 를 입력하고, ActionEntity 에는 각각 Spricket_R 과 Sprocket_K 을 입력합니다.



| Header_JointEx_Revolute | Name | BaseEntity | ActionEntity | RefFrame |
|-------------------------|----------------|---------------|--------------|---------------|
| JointEx_Revolute | Rev_Sprocket_R | Track_Frame_R | Sprocket_R | HC_Sprocket_R |
| JointEx_Revolute | Rev_Sprocket_L | Track_Frame_L | Sprocket_L | HC_Sprocket_L |

3. 이 때 각 Revolute Joint 에 모션을 정의하는 데, 이는 옵션을 이용하여 입력합니다.



| UseMotion | Motion |
|-----------|---------------|
| TRUE | Info_Joint_01 |

TRUE Info_Joint_01

- 이 때, Info_Joint_01 이라는 Motion 옵션을 사용하기 위해서는 아래와 같이 Motion 을 정의해줘야 합니다.
- 우선 Motion 에 사용할 Expression 을 정의합니다.



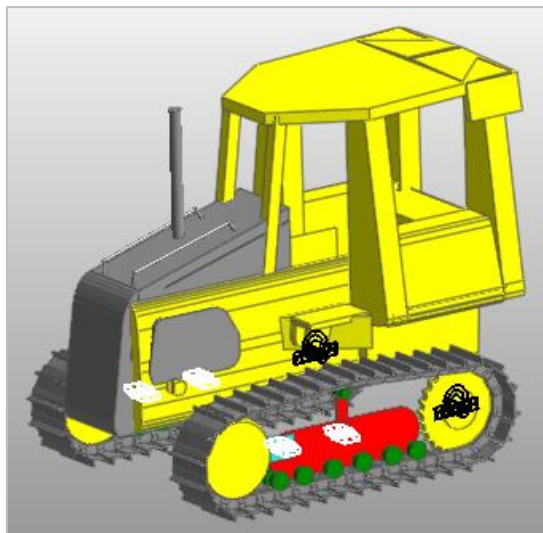
| Header_SubEntity_Expression | Name | ExpressionText |
|-----------------------------|------------------|------------------------------|
| SubEntity_Expression | Exp_Sprocket_Vel | -STEP(TIME, 0.1, 0, 1, 360D) |

- 아래와 같이 Info_Joint_01 이라는 이름의 옵션을 입력하고 사용할 Expression 의 이름과 MotionType, MotionCondition 을 입력합니다.



| Header_Info_JointMotion | Name | Expression | MotionType | MotionCondition |
|-------------------------|---------------|------------------|------------|-----------------|
| Info_JointMotion | Info_Joint_01 | Exp_Sprocket_Vel | Standard | Velocity |

- 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이 생성됩니다. 이 때, 생성된 Joint 는 다음과 같습니다.



Track Frame 과 **Track Body** 사이의 **Revolute Joint** 생성:

1. Frame Body 에 각 Track Body 를 Revolute Joint 로 고정합니다.
2. 아래와 같이 우측의 언더캐리지 쪽의 Track Frame 과 Wheel 을 연결합니다.



| Header_JointEx_Revolute | Name | BaseEntity | ActionEntity | RefFrame |
|-------------------------|---------------|---------------|---------------|--------------|
| JointEx_Revolute | Rev_Wheel_1_R | Track_Frame_R | RoadWheel_1_R | HC_Wheel_1_R |
| JointEx_Revolute | Rev_Wheel_2_R | Track_Frame_R | RoadWheel_2_R | HC_Wheel_2_R |
| JointEx_Revolute | Rev_Wheel_3_R | Track_Frame_R | RoadWheel_3_R | HC_Wheel_3_R |
| JointEx_Revolute | Rev_Wheel_4_R | Track_Frame_R | RoadWheel_4_R | HC_Wheel_4_R |
| JointEx_Revolute | Rev_Wheel_5_R | Track_Frame_R | RoadWheel_5_R | HC_Wheel_5_R |
| JointEx_Revolute | Rev_Wheel_6_R | Track_Frame_R | RoadWheel_6_R | HC_Wheel_6_R |

3. 이번에는 아래와 같이 Track Frame 과 Carrier 을 연결합니다.



| Header_JointEx_Revolute | Name | BaseEntity | ActionEntity | RefFrame |
|-------------------------|----------------------|---------------|--------------|-----------------------|
| JointEx_Revolute | Rev_FlangeCenter_1_R | Track_Frame_R | Carrier_R | HC_Carrier_Roller_1_R |

4. 이번에는 아래와 같이 Idler 와 Tentioner 를 연결합니다.



| Header_JointEx_Revolute | Name | BaseEntity | ActionEntity | RefFrame |
|-------------------------|-------------|------------|--------------|------------|
| JointEx_Revolute | Rev_Idler_R | Idler_R | Tensioner_R | HC_Idler_R |

5. 왼쪽 언더캐리지의 Track Frame 과 Wheel 도 아래와 같이 연결합니다.



| Header_JointEx_Revolute | Name | BaseEntity | ActionEntity | RefFrame |
|-------------------------|----------------------|---------------|---------------|-----------------------|
| JointEx_Revolute | Rev_Wheel_1_L | Track_Frame_L | RoadWheel_1_L | HC_Wheel_1_L |
| JointEx_Revolute | Rev_Wheel_2_L | Track_Frame_L | RoadWheel_2_L | HC_Wheel_2_L |
| JointEx_Revolute | Rev_Wheel_3_L | Track_Frame_L | RoadWheel_3_L | HC_Wheel_3_L |
| JointEx_Revolute | Rev_Wheel_4_L | Track_Frame_L | RoadWheel_4_L | HC_Wheel_4_L |
| JointEx_Revolute | Rev_Wheel_5_L | Track_Frame_L | RoadWheel_5_L | HC_Wheel_5_L |
| JointEx_Revolute | Rev_Wheel_6_L | Track_Frame_L | RoadWheel_6_L | HC_Wheel_6_L |
| JointEx_Revolute | Rev_FlangeCenter_1_L | Track_Frame_L | Carrier_L | HC_Carrier_Roller_1_L |

JointEx_Revolute

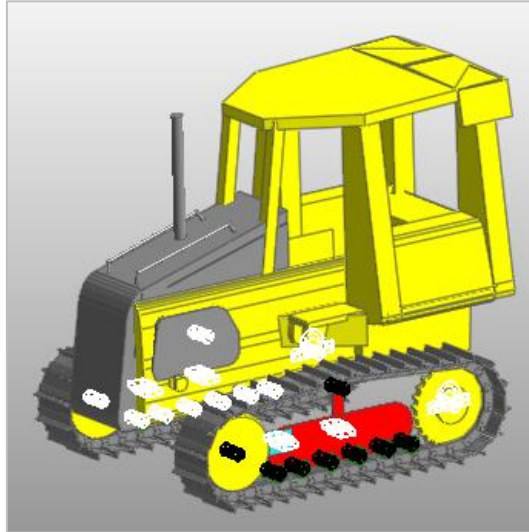
Rev_Idler_L

Idler_L

Tensioner_L

HC_Idler_L

6. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같은 모델이



생성됩니다. 이 때, 생성된 Joint 는 다음과 같습니다.

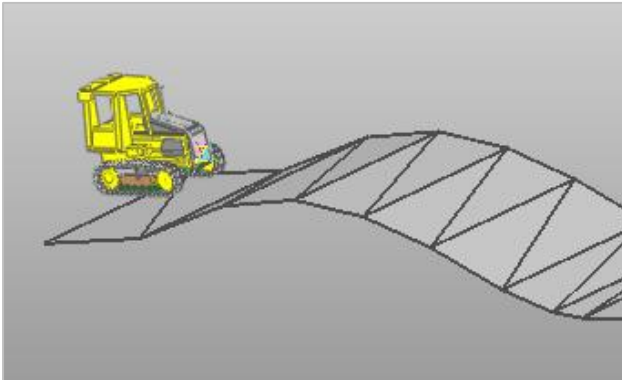
이리하여 Track System 의 전체 모델링이 완료되었습니다.

Ground 생성

1. 이번에는 앞서 생성한 Track System 이 움직일 '로드(road)'를 Import 하도록 합니다.
2. 튜토리얼과 함께 제공된 **Terrain_Outline.rdf** 을 Import 하기 위해 아래의 내용을 엑셀에 입력합니다. 다른 로드를 사용하고 싶다면 import 할 rdf 파일만 변경해주면 됩니다.



| Header_RoadData_Import | Name | FileName |
|------------------------|-----------------|---------------------|
| RoadData_Import | Terrain_Outline | Terrain_Outline.rdf |



3. 엑셀파일을 저장하고 Run 을 이용하여 엑셀파일을 읽어 들이면 아래와 같이 로드가 생성됩니다.

Chapter

8

eTemplate 을 이용한 Setting 변경하기

목적

이 장에서는 다음과 같이 몇 가지 Setting 을 설정하게 될 것입니다.

- Dynamic Analysis Setting
- Icon Size 및 Marker Size 변경
- 작업 공간 색상 변경



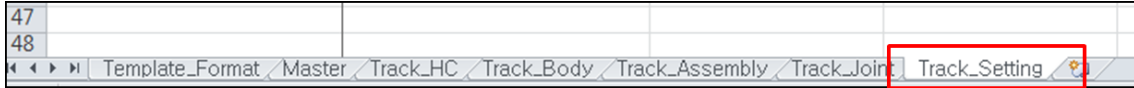
예상 소요 시간

10 분

Dynamic Analysis Setting 변경

Dynamic Analysis 설정:

1. 지금부터는 **Track_Setting** Sheet 에 내용을 입력합니다



2. 해석을 위한 Simulation Time (End Time), Simulation Step (Step), Plot Multiplier Step Factor 값을 eTemplate 에서 입력할 수 있습니다.
3. 아래와 같이 Simulation Time 은 10 초, Simulation Step 은 250, Plot Multiplier Step Factor 는 1 으로 설정하겠습니다.



| Header_Setting_DynamicAnalysis | SimulationTime | SimulationStep | PlotMultiplierStepFactor |
|--------------------------------|----------------|----------------|--------------------------|
| Setting_DynamicAnalysis | 10 | 250 | 1 |

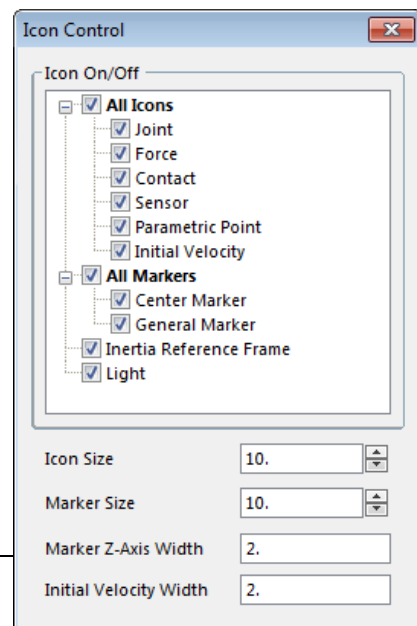
Display Setting 변경

Icon Setting 설정:

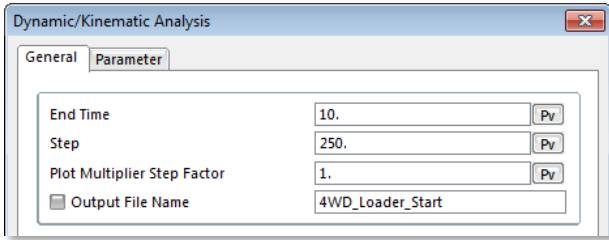
1. eTemplate 을 이용하여 화면에서 보이는 아이콘에 대해서도 설정이 가능합니다.
2. 여기서는 Icon 의 크기와 Marker 의 크기를 변경해 보겠습니다.
3. Icon 크기와 Marker 크기는 10 으로 설정하겠습니다.



| Header_Setting_Icon | IconSize | MarkerSize |
|---------------------|----------|------------|
| Setting_Icon | 10 | 10 |



4. 위에서 한 설정은 아래와 동일합니다.



BackGround Setting 설정:

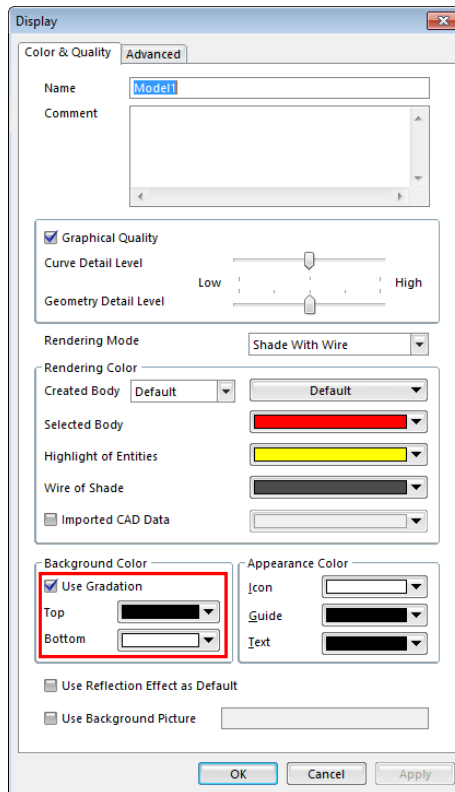
1. eTemplate 을 이용하여 화면의 배경 색상을 변경할 수 있습니다.
2. 아래와 같이 Gradation 을 이용한 설정을 합니다.



| | | | |
|---------------------------|-------------|----------|--------------|
| Header_Setting_BackGround | BottomColor | TopColor | UseGradation |
| Setting_BackGround | White | Black | TRUE |

3. 제공되는 Color 는 eTemplate Manual 에 제공되는 Appendix 를 참고해주세요. (아래는 RecurDyn V8R2 기준으로 지원되고 있는 Color 입니다. 물론 RecurDyn 의 GUI 를 통해서는

| | | |
|---|-------------|--|
| | C | |
| e | Color | |
| | Red | |
| | Green | |
| | Blue | |
| | Yellow | |
| | Pink | |
| | Brown | |
| | GreenYellow | |
| | Azure | |
| | White | |
| | SpringGreen | |
| | Black | |
| | Gray | |



보다 다양한 Color 의 입력이 가능합니다.)

Tip: 좌표입력 방식과 참조 입력 방식

지금까지 여러 개의 Sheet 에 데이터를 나누어 입력했습니다. 하지만 이는 데이터를 쉽게 구분하기 위한 목적이었으며, 실제로는 모든 내용을 Master Sheet 에 입력해도 동일하게 동작합니다. Master Sheet 를 이용할 경우, Chapter 3 에서 진행했던 어떤 Sheet 를 활용할 지 여부에 대해서는 정보를 입력하지 않아도 되며, 그 때는 사용하지 않는 Sheet 는 모두 지워도 무방합니다.

eTemplate 파일의 import 및 해석 수행하기

목적

이 장에서는 다음과 같이 몇 가지 과정을 거치게 될 것입니다.

- eTemplate 파일의 Import
- 해석 수행



예상 소요 시간

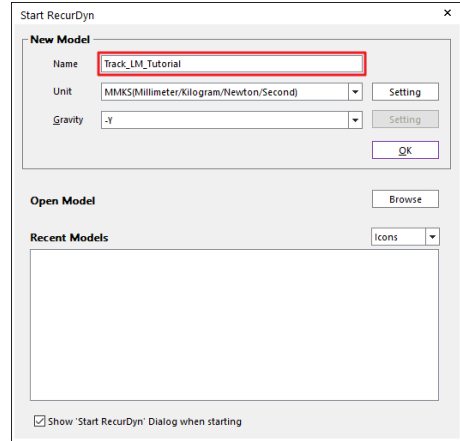
10 분

eTemplate 파일의 Import

eTemplate Import 를 위한 새로운 모델 생성하기:

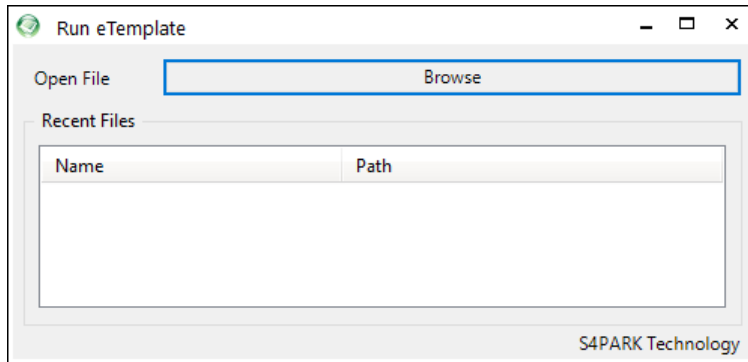
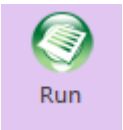
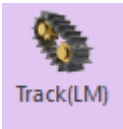


1. 바탕화면에서 **RecurDyn** 아이콘을 클릭합니다.
2. Model Name 입력란에 새 모델의 이름을 **Track_LM_Tutorial** 으로 입력합니다.

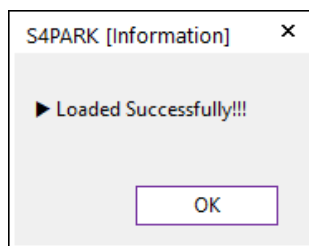


eTemplate 실행:

1. ModuleKey 설정이 **S4PARK_Module_TrackLM** 으로 설정되어 있으므로, Track_LM Subsystem 으로 들어가야만 합니다.
2. **Toolkit** 탭의 **Subsystem Toolkit** 그룹에서 **Track(LM)**을 클릭합니다.
3. **Customize** 탭의 eTemplate 그룹에서 **Run Icon** 을 클릭합니다.
그러면 다음과 같이 Run eTemplate 다이얼로그가 나타납니다.



4. Browse 버튼을 클릭하여, 앞서 생성한 엑셀파일을 Import 합니다. 이때, 엑셀파일과 본 튜토리얼에서 사용되는 파일들 (*.x_t, *.mat 등) 은 동일한 폴더에 있어야 합니다.
5. 잠시 기다리면, 모델 생성이 완료되었다는 메시지가 나옵니다.

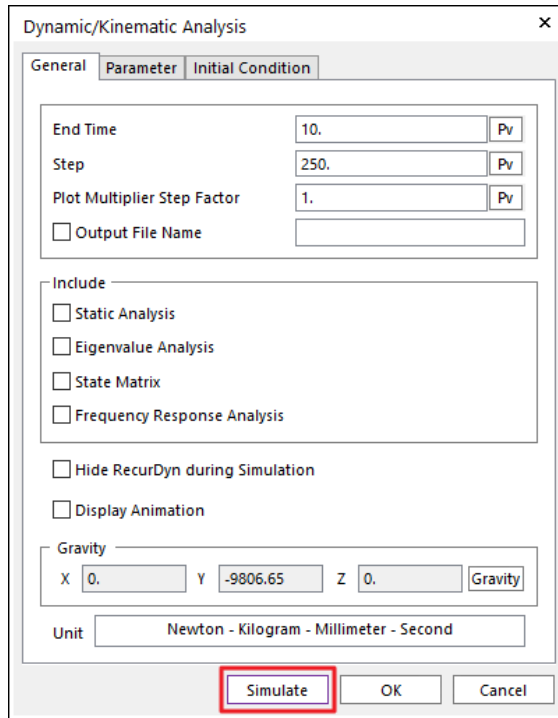


해석 수행

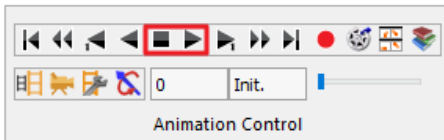
Track_LM 해석:



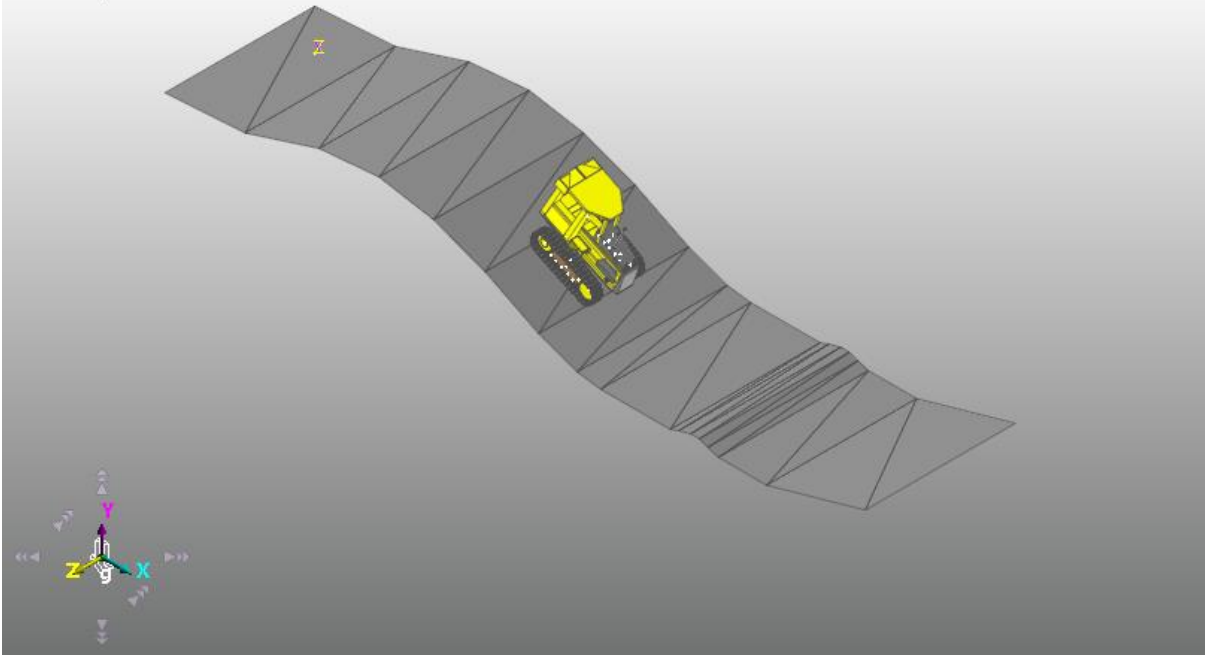
1. **Analysis** 탭의 **Simulation Type** 그룹에 있는 **Dyn/Kin** 아이콘을 클릭합니다. Simulation 을 실행합니다.
2. Chapter 7 에서 입력한 대로 값이 입력되어 있는 것을 확인할 수 있습니다.
3. **Simulation** Button 을 클릭하여 해석을 시작합니다.



4. 해석을 완료하고 Simulation 을 확인합니다.



TrackLM_1@Model1
Time = 6,84000000 Second



Thanks for participating in this tutorial