

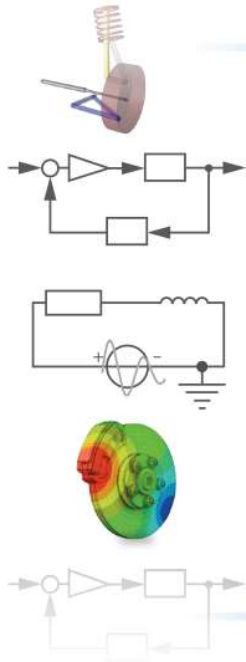


DEMOBASE

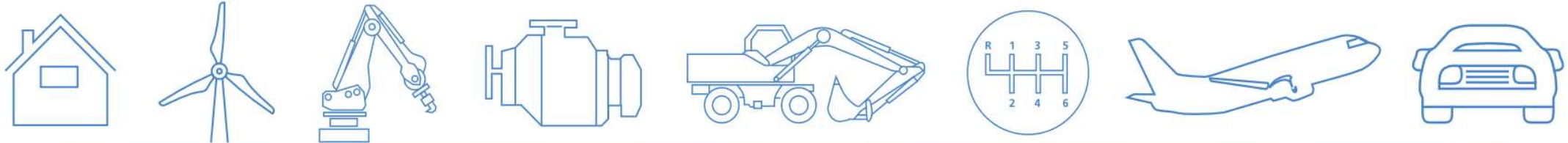
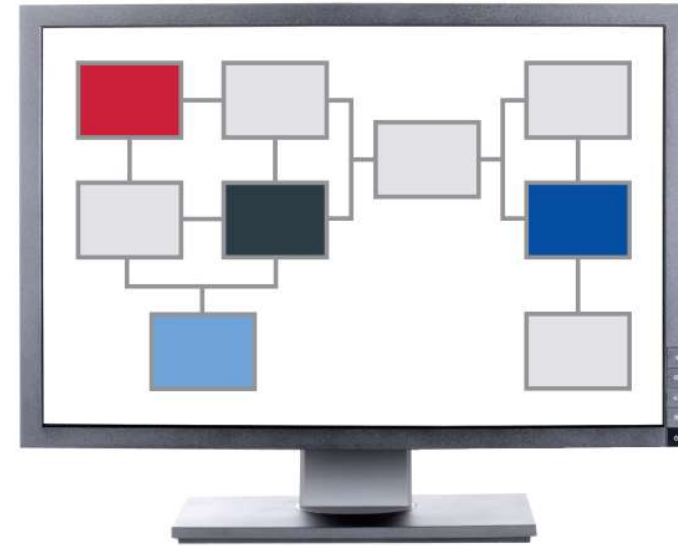
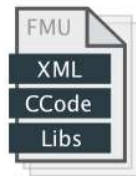
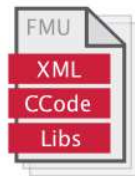
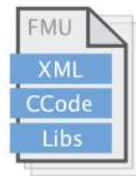
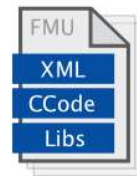
COLLABORATIVE PLATFORM FOR INTEGRATING MODELS WITH FMI

Thomas Fickenscher – MODELON AB 2020-09-16










Modelon

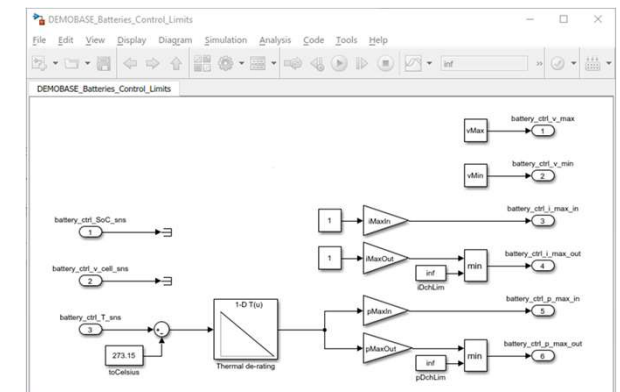
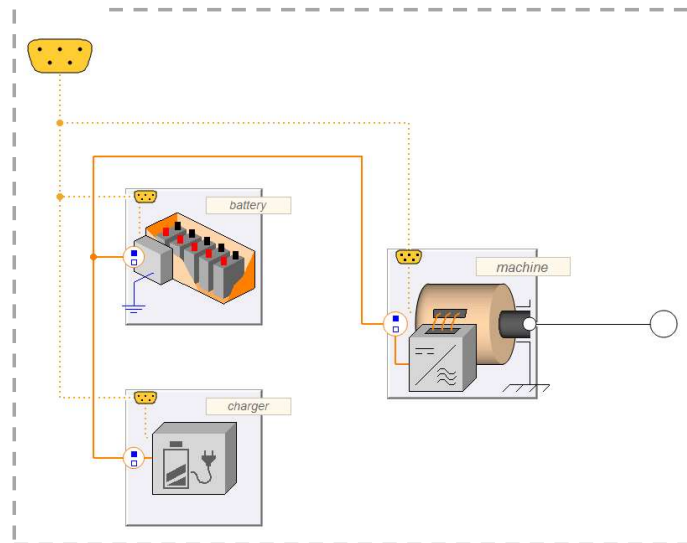


fm: Functional Mock-Up Interface

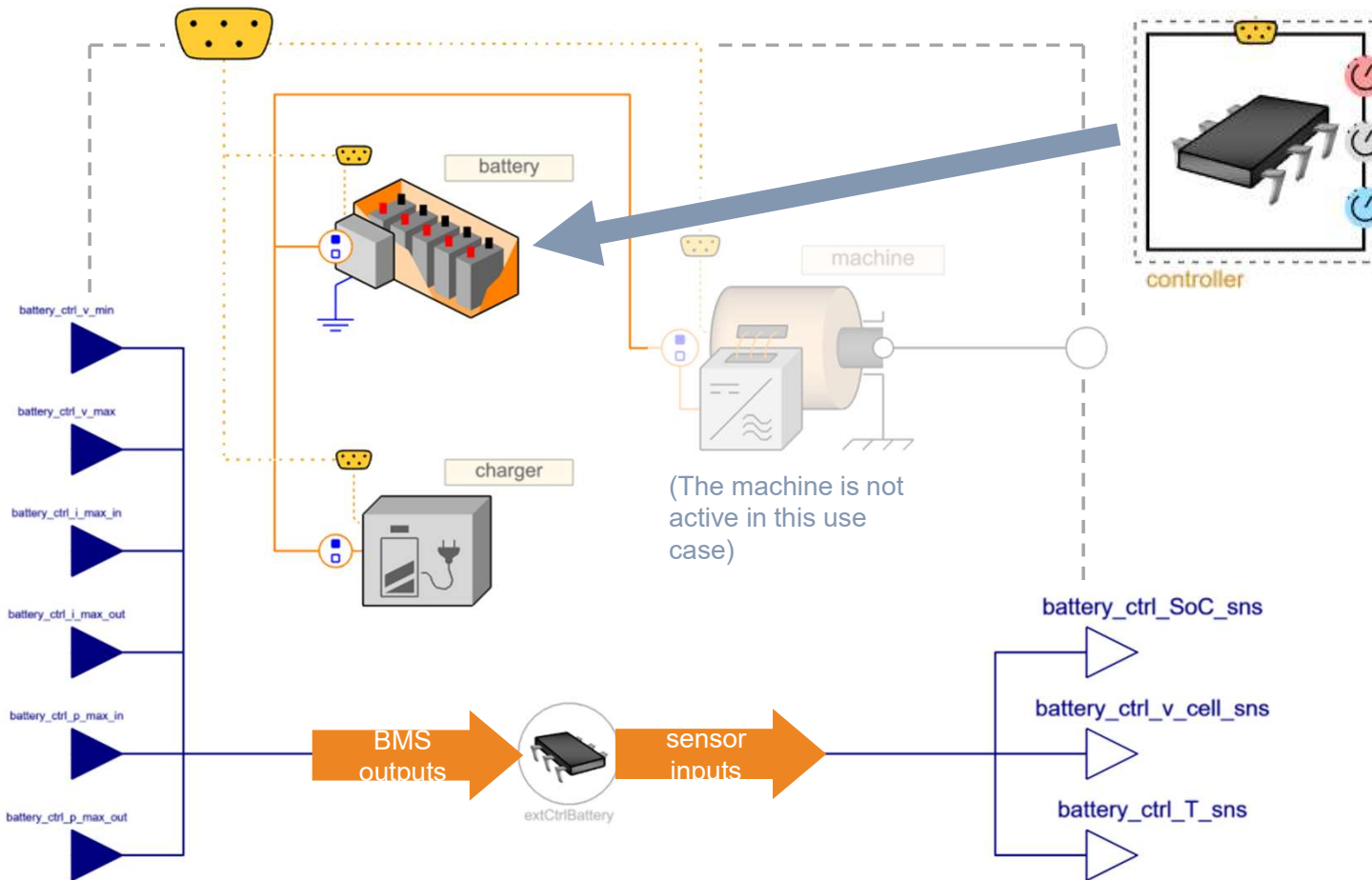


CASE STUDY : COMBINE AN ELECTRIC PROPULSION SYSTEM MODEL FMU WITH AN EXTERNAL BMS CONTROLLER FMU INTO A SINGLE SYSTEM FMU

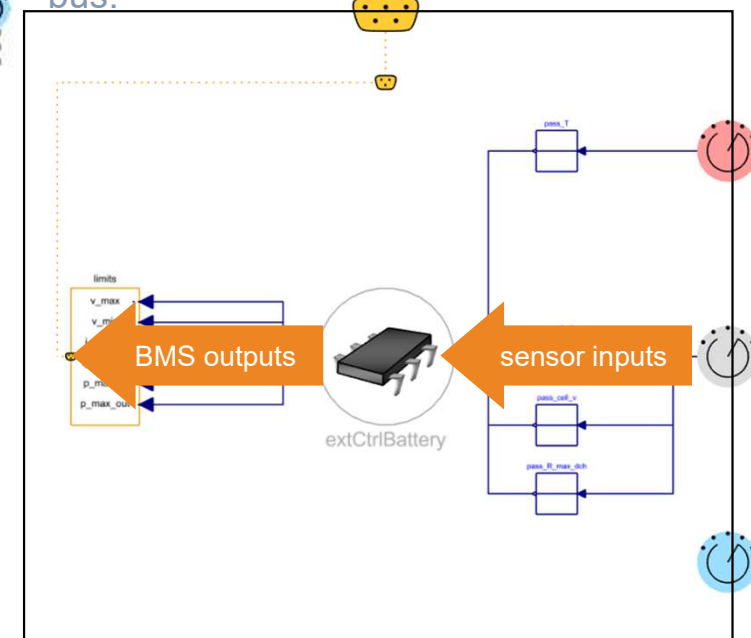
- ▶  Electrification
- ▶  Modelon
- ▶  DEMOBASE 
- ▶  Systems
- ▶  Batteries
- ▶  Machines
- ▶  Loads
- ▶  Converters



USE CASE – PLANT SYSTEM FMU + EXTERNAL BMS FMU



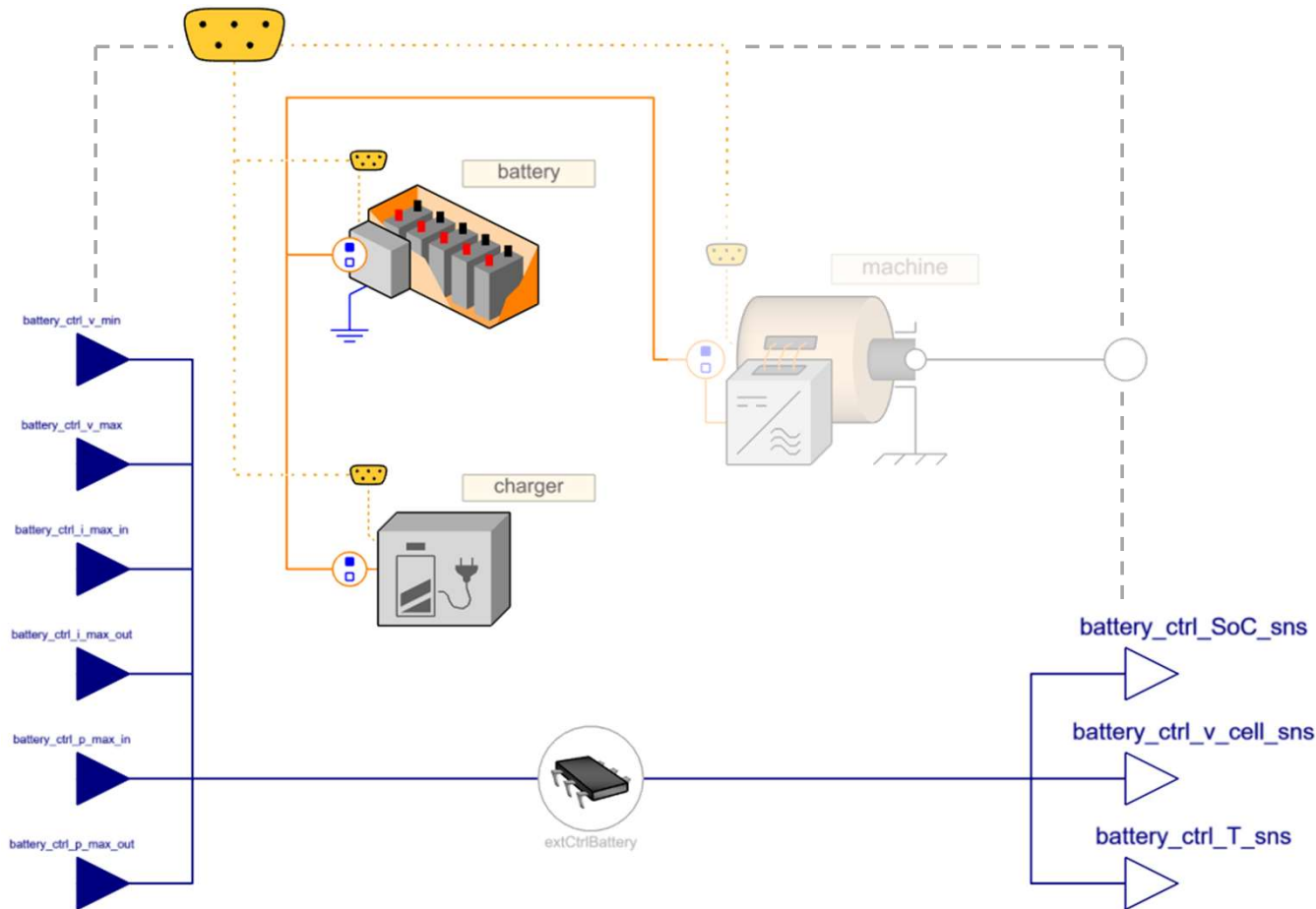
“empty” battery controller
Forwards sensor inputs to external BMS,
and returns BMS outputs to the control
bus.



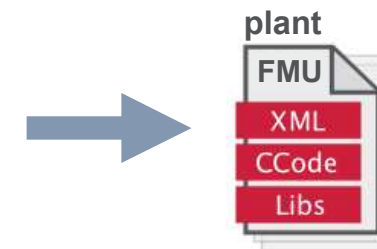
USE CASE – PLANT SYSTEM FMU + EXTERNAL BMS FMU



Modelon Impact

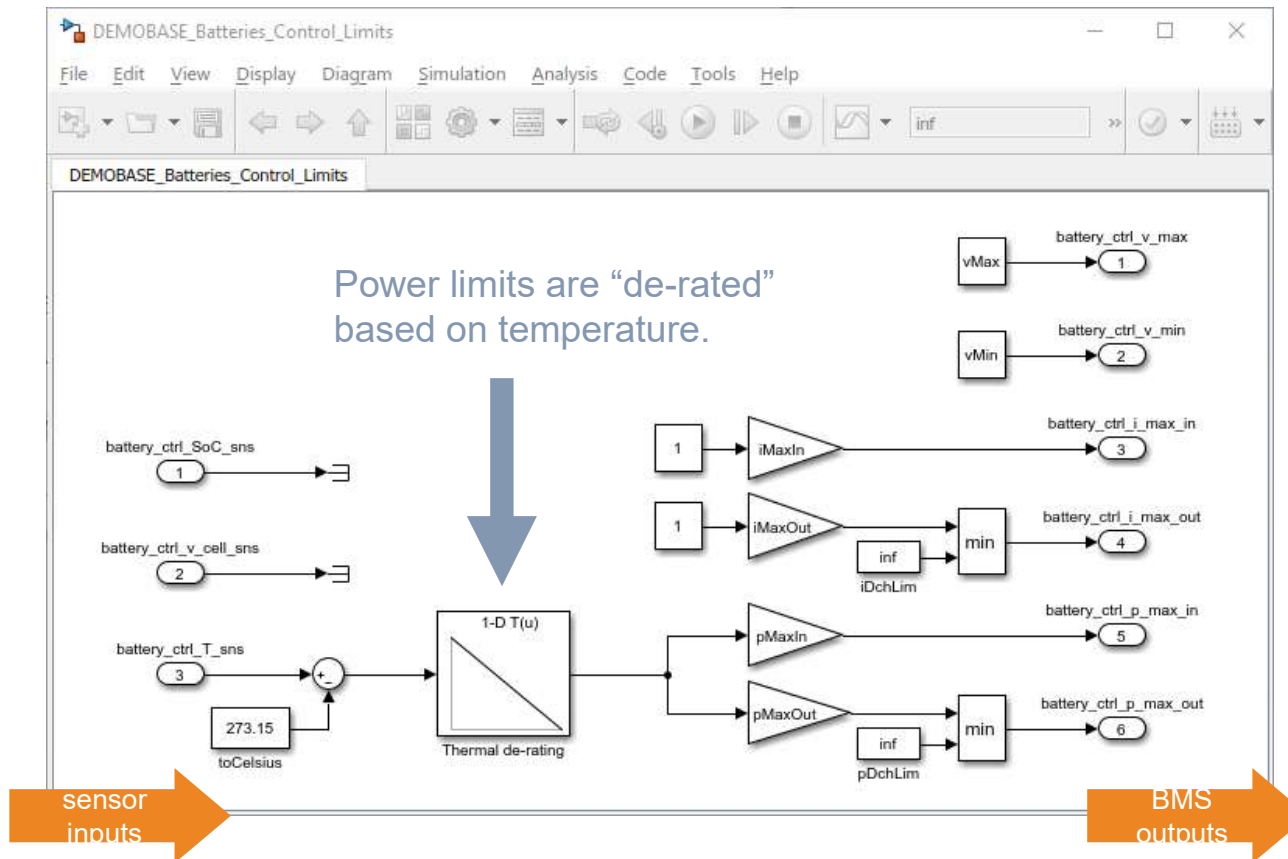


©2020 Modelon. All Rights Reserved.



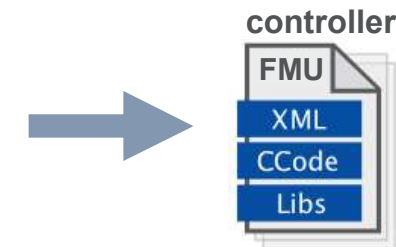
FMU is generated from plant model
in Modelon Impact

USE CASE – PLANT SYSTEM FMU + EXTERNAL BMS FMU



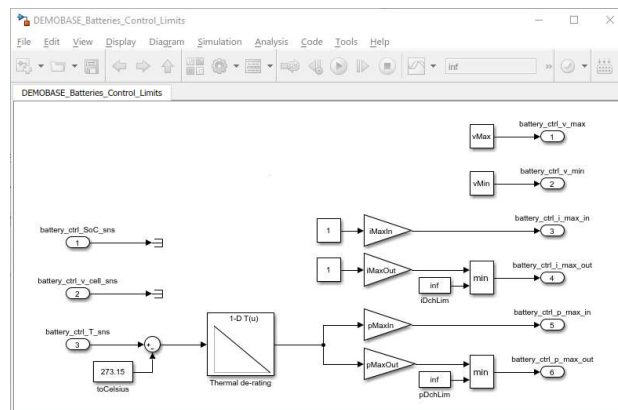
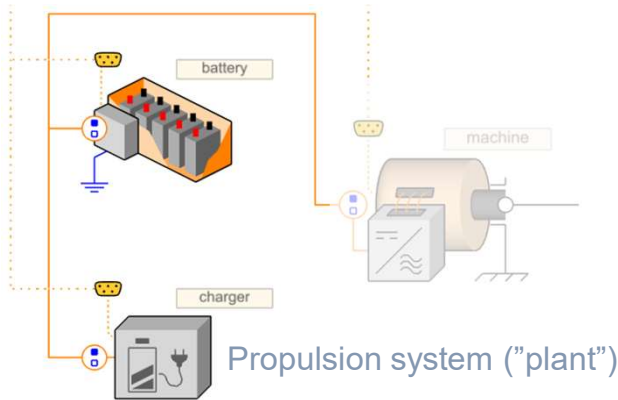
Simulink battery controller

Outputs: battery limits (voltage, current, power)
Inputs: SoC, cell voltages, temperatures



Exported as FMU

(using the Modelon FMI Toolbox for Simulink)

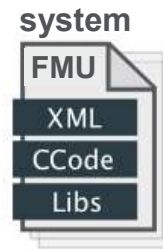


BMS ("controller")



Integrate to system FMU ...

FMI Composer backend



Simulate system ...

PyFMI

```
In [3]: system_name = "ElectricPropulsionSystemCharging"
system_folder = './' + system_name
system_fmu = system_name + ".fmu"

system = load_exp(exp_path = system_folder)
options = FMUExportOptions(FMUKind.PC)
options['blockout'] = True
fmu = system.export_fmu(generate_to = system_fmu, options=options)

##### Simulate #####

root : DEBUG SystemStructure.xml specifies SSP standard version "1.0".
root : INFO Loading "BatteryController" FMU...
root : INFO XML specifies FMI standard version 2.0
root : INFO Successfully loaded FMU
root : INFO Loading "ElectricPropulsionSystem" FMU...
root : INFO XML specifies FMI standard version 2.0
root : INFO Successfully loaded FMU
root : DEBUG resources/root.xsv specifies SSP standard version "1.0".
root : DEBUG resources/root.xsv specifies SSP standard version "1.0".
root : INFO Creating PE aggregate
root : INFO Treating all FMIs of kind "Combined" as PE
root : INFO FMU successfully exported to "ElectricPropulsionSystemCharging.fmu"

In [4]: from pyfmi import load_fmu

model = load_fmu(system_name=".fmu")
model.set("ElectricPropulsionSystem.battery_soc_start",0.2)

Solver options:

Solver          : Cvxode
Linear multistep method : QSF
Nonlinear solver  : Newton
Linear solver type : DENSE
Maximal order    : 5
Tolerances (absolute) : 1e-05
Tolerances (relative) : 0.0001

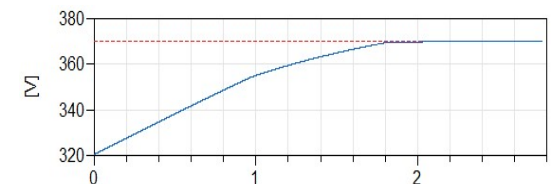
Simulation interval : 0.0 - 2000.0 seconds.
Elapsed simulation time: 3.0501292 seconds.

In [5]: from matplotlib import pyplot as plt
import matplotlib inline

figSoC = plt.figure()
ax = figSoC.add_subplot(1, 1, 1)
ax.plot(res['time'], res['SoC'])

figT = plt.figure()
ax = figT.add_subplot(1, 1, 1)
ax.plot(res['time'], res['T']) - 273.15)
plt.show()

Out[5]: (10, 30)
```





Modelon Impact

System Simulation Software for Everyone
Designed for Collaboration
Accelerate Decision Making With
Confidence

©2020 Modelon. All Rights Reserved.



THANK YOU



Thomas.Fickenscher@modelon.com

Tel: +44 77 8494 3456