Model information and Executing the Execution_main script

- Four versions of the model have been distributed as part of the releasable package; one for each of summer, winter, autumn and spring in the baselined year (2021/22). The LV and general loads in each of the four seasonal versions of the model are scaled differently to represent the daily load and solar generation profiles of those seasons. Note that the variable “Season” on line 35 of the Execution_main script must be set to one of the four strings “Summer”, “Winter”, “Autumn” or “Spring” as per the version of the model that is being utilised.
- The model is baselined to present-day conditions. It will reproduce present-day power flows and voltages for typical high or low solar conditions during any of the four seasons of the year.
- The model can be configured to run for high or low solar conditions using by setting the variable “SolarOn” on line 36 of the script to 1 or 0 respectively.
- The model is configured for present-day levels of rooftop solar photovoltaic (PV) generation and electric vehicles (EVs).
- To add future amounts of PV to the model, run either one of the scripts. “PV_grower_residential” or “PV_grower_CnI”; these will add residential or commercial/industrial PV of a given desired amount respectively.
- To add EVs to the model, run the script “EV_grower” on it.
- The Execution_main script also includes a variable “QDSLsOn” on line 37 which switches the volt-watt and volt-var algorithms on and off if it is set to 1 or 0 respectively. It is advised that this parameter is set to 1 in general for realism.
- A subset of the PV generators in the model have their QDSL component settings configured to implement volt-watt but not implemented volt-var. Another subset of them do not have QDSL components assigned to them. These subsets both exist because they correspond to PV installations that were commissioned in years during which the Australian standard AS4777.2 did not include requirements for the volt-watt or volt-var algorithms to be implemented. When the PV_grower scripts are executed on the model however, QDSL components with both volt-watt and volt-var functions active are gradually added to these inverters; this corresponds to the assumption that future PV installations will exhibit universal adherence to the current revision of AS4777.2.
- The Execution_main script also includes a variable “EVsOn” on line 38 which activates EV charging when set to 1 and deactivates them when set to 0. It is advised that this parameter is set to 1 in general for realism.
- Different sections of code should be uncommented depending on whether all half-hour periods across a whole day are to be simulated or only a single half-hour period is to be simulated. For the former, un-comment lines 113-114 and comment-out lines 117-119. To simulate only the hour centred on noon, comment-out lines 113-114 and un-comment lines 117-119. To simulate another time-period across the day, alter the number that appears on each of lines 117 and 119 to a different value (e.g. 24 corresponds to noon, 36 corresponds to 6pm and 12 corresponds to 6am).
- Line 178 of Execution_main implements an initial guess at the voltage that is furnished at the zone substation MV bus due to the LDC algorithm that is implemented there; an approximate representation of the LDC algorithm is then implemented on lines 186-200. When Executing simulations across the entire day, comment line 178 of Execution_main out for faster performance.

- For faster performance, leave line 178 of Execution_main un-commented when Executing a simulation at only one time of day. When Executing simulations only at noon under high solar conditions or at 6pm-8pm during the evening peak, 0.9875pu and 1.0125pu are generally accurate guesses at the outcome of the zone substation LDC algorithm that can be applied to line 178 of Execution_main.
- To make Execution_main produce an Excel file with bus voltage results for each time of day simulated, ensure that lines 225 and 228 are un-commented.
- To make Execution_main produce an Excel file with transformer thermal utilisation results for each time of day simulated, ensure that lines 226 and 229 are un-commented.
- To make Execution_main produce an Excel file with distribution line and buried cable thermal utilisation results for each time of day simulated, ensure that lines 227 and 230 are un-commented.
- Lines 104 and 107 can be un-commented to enable a function which will place STATCOMs at all of the buses for which strings representing their name are entered into the list on line 104. Note that this function will permanently add new STATCOMs at each of the listed buses on every occasion that the Execution_main script is run with these lines un-commented; so if additional runs with the same configurations of STATCOMs are desired then the same instance of the model should have Execution_main run on it again with these lines commented out. Note also that the function "realise_STATCOMs()" is configured to apply particular V-Q droop values and voltage thresholds to the new STATCOMs; alteration of this function inside the script "exec_utils" is necessary if these parameters are to be changed.

Executing the PV_grower_residential and PV_grower_CnI scripts

- To configure the amount of residential rooftop PV capacity that the model will be projected to, set the variable "target_residential_avg_PV_size" on line 25 of PV_grower_residential to the desired average kVA of installed PV per household. Note that the average amount of residential PV inverter capacity per household in the baselined version of the model is approximately 1.9kW; this variable must therefore be set to a value higher than 1.9 if any changes are to be made using PV_grower_residential.
- To configure the amount of commercial/industrial rooftop PV capacity that the model will be projected to, set the variable "target_CnI_avg_PV_size" on line 25 of PV_grower_CnI to the desired average kVA of installed PV per commercial/industrial customer. Note that the average amount of commercial/industrial PV inverter capacity per customer in the baselined version of the model is approximately 2.56kW; this variable must therefore be set to a value higher than 2.56 if any changes are to be made using PV_grower_CnI.

Executing the EV_grower script

- Once EV_grower has been run on the model, the names of the LV loads will be appended with several smaller letters to indicate encoding of EV ownership by the residential customers represented in the model; don't be alarmed this is not an error
- The EVs encoded in the model won't be made to draw additional load from the model unless the variable "EVs_On" on line 81 of Execution_main is set to 1
- If Execution_main is run with EVs_On set to 1 and it crashes midway through for any reason, the LV load sizes will be configured incorrectly. If this occurs, the original seasonal version of the model from the releasable package must be re-loaded into PowerFactory for the load and solar generation scaling factors to be configured correctly.

- The variable “target_residential_avg_EV_count” on line 22 of EV_grower must be set to a value greater than or equal to 0 to specify the desired average amount of EVs per household that the model will be projected to.

Executing the Conductor_upgrade script

- If investigation of the effect of re-conductoring and/or distribution transformer upgrades in the model is desired, specific individual overhead lines (OHLs), buried cables or distribution transformers or OHLs or buried cables in selections of areas can be upgraded by executing the script “Conductor_upgrade”.
- To configure which areas will see their OHLs and/or buried cables upgraded to the next available rating using this script, enter valid strings into any of the arrays on lines 63-68.
- Enter strings corresponding to the names of specific transformer-substations into the array “LV_feeder_set_all” to enact upgrades on all of the OHLs and buried cables in the corresponding LV areas within each of these transformer-substations.
- Enter strings corresponding to the names of specific transformer-substations into the array “LV_feeder_set_cable” to enact upgrades on only the buried cables in the corresponding LV areas within each of these transformer-substations.
- Enter strings corresponding to the names of specific transformer-substations into the array “LV_feeder_set_OHL” to enact upgrades on only the OHLs in the corresponding LV areas within each of these transformer-substations.
- Enter strings corresponding to the names of specific individual LV buried cables or LV OHLs in the model which are to be upgraded in the array “LV_line_string_set”.
- Enter strings corresponding to the names of specific individual MV buried cables or MV OHLs in the model which are to be upgraded in the array “MV_line_string_set”.
- Enter strings corresponding to the names of specific individual distribution transformers in the model which are to be upgraded in the array “transformers_to_be_upgraded”.