



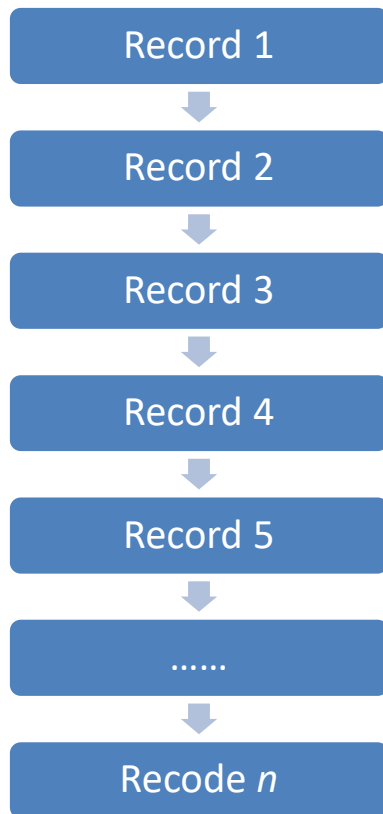
ASM Mini Sharing

# DATA MANIPULATION USING DCS (09.04.2019)

# Agenda

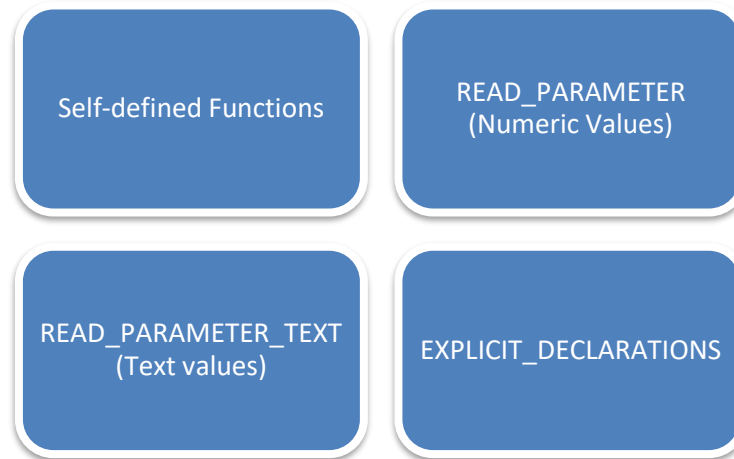
- How does DCS work generally?
- Approach I: Use data from two systems
- Approach II: Combine account balance data from separate files
- Approach III: Combine basic & rider in one record

# How does DCS Work Generally?



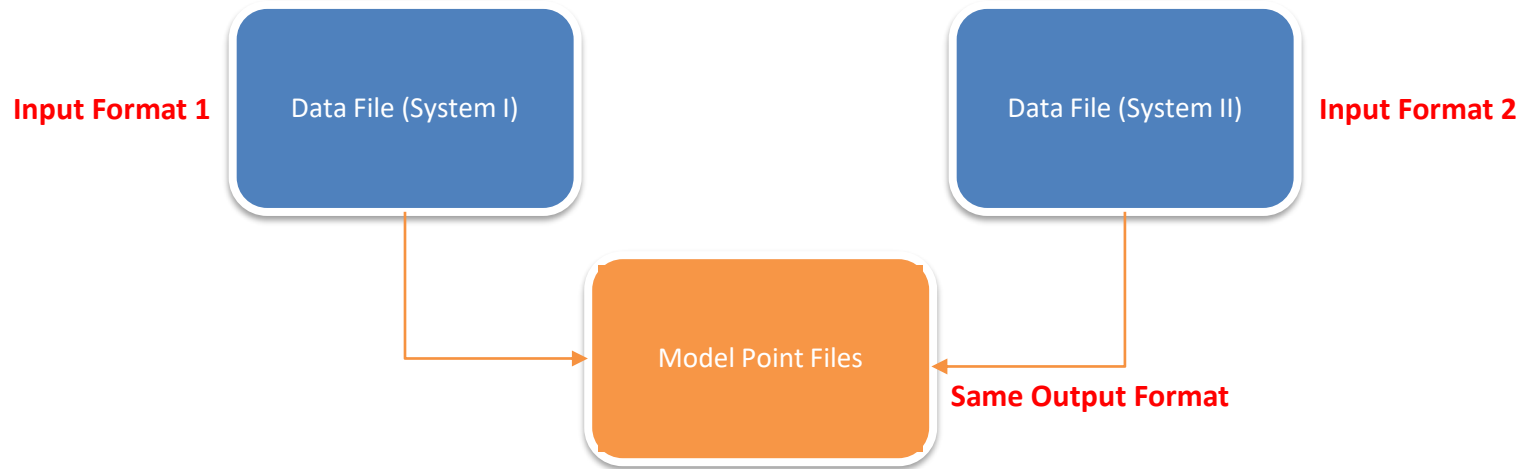
- **Data Conversion System** (“DSC”) is a supplementary application that comes together with Prophet. The main use of DCS is to produce model point files (“MPF”) that can be fed into actuarial models designed in Prophet.
  - Generally, a MPF is a text file delimited with comma (ignoring the codes after the last data record).
  - MPF can use any file extensions that we prefer. The common file extensions are “.NB” for new business processing MPF & “.IF” for in force runs.
  - We can use other applications to produce MPF, as long as they can create text files delimited with comma.
- Similar to how Prophet works, DCS processes data files “record-by-record” – i.e. DCS will complete the entire process (everything you have written as codes!) for a particular record before moving to the next record.
  - The concept may be different from other data manipulation applications, which may also process data by “**tables**”.
- The process includes reading values from the input files (automatically, you do not need to have extra codes for this step), perform validations & calculations and generate MPF (by using `OUTPUT`). MPFs are produced after DCS has completed processing all records from the input files.

# Useful Techniques



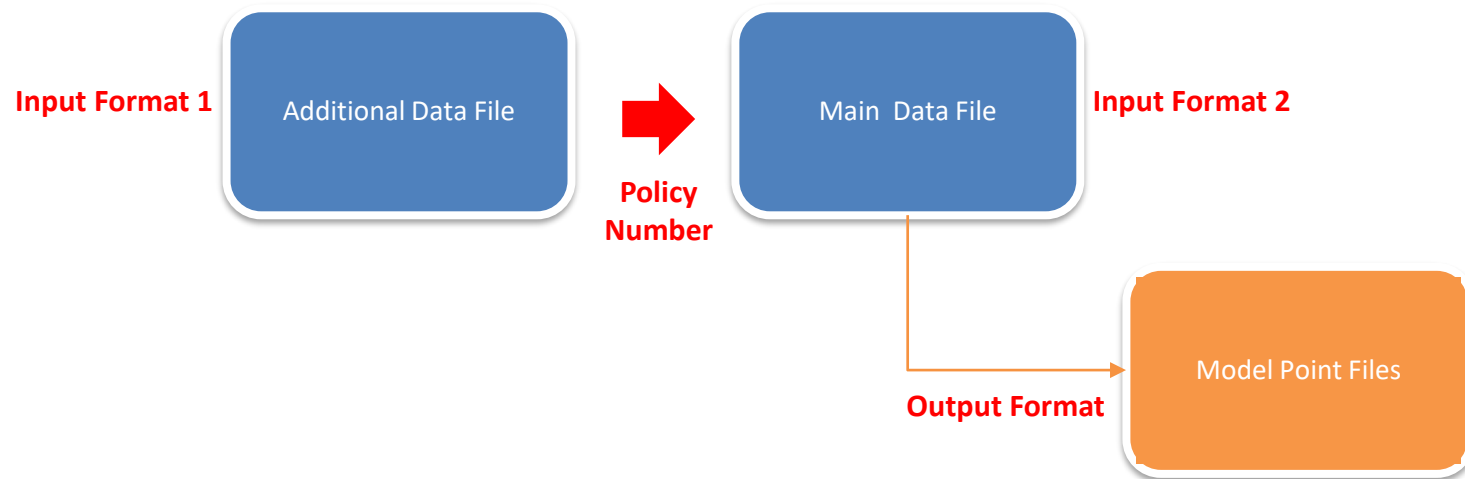
- **Self-defined Functions** – For formula sets that are used repeated at different points of main codes.
  - Should be placed before the main codes.
  - Coding to be included between `FUNCTION ... ENDFUNCTION`.
  - Indicate the variable which the value should be returned to the main codes, by using `RETURN <var_name>`.
- **READ\_PARAMETER & READ\_PARAMETER\_TEXT** – Allow users to enter run parameters directly in a pop-up dialog, without needing to amend the Run Settings. This is useful to setup a DCS program that can be reused without requiring any amendments.
- **EXPLICIT\_DECLARATIONS** – Allow DCS to run faster. It also helps to reduce the likelihood of coding errors. Should be placed as the first line of coding (ignoring the comments) after functions.

# Approach I: Use Data from Two Systems



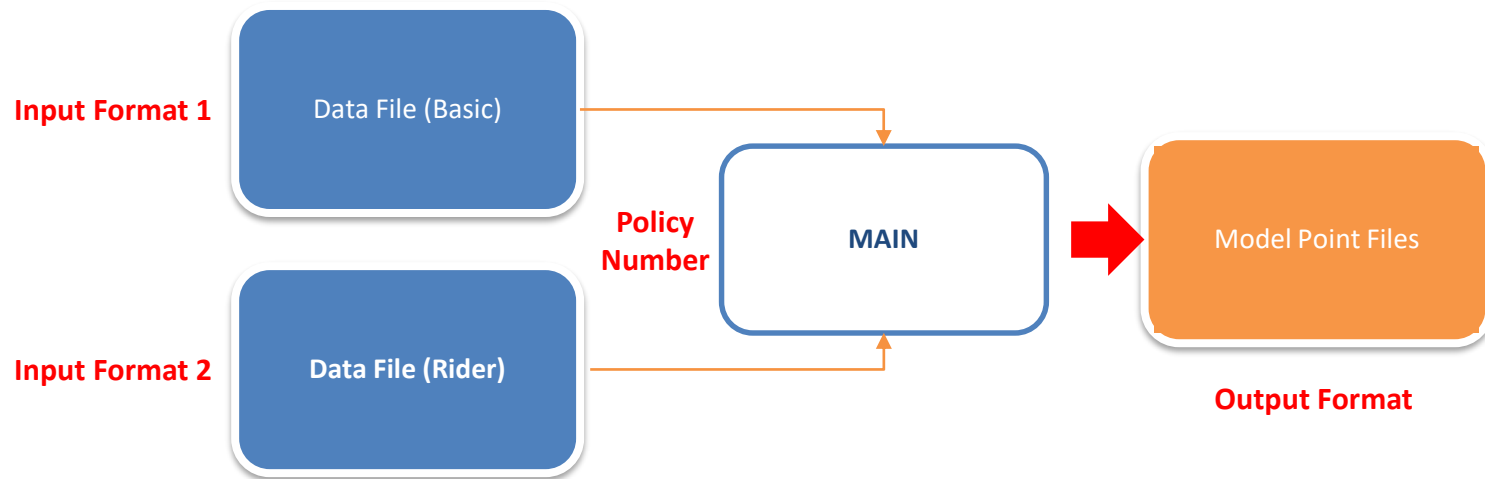
- If the data files have the same structure, we can use wildcards (“\*”) to specify the file names; if the data files have different structures, we need to define different input formats.
- Both input formats should be classified as “Main Records”. It is important to note that both data files should be the same types, e.g. using the same field delimiter.
- The recommended field delimiter for text files is **TAB**, as it is considered as the “safest” option – since it is not likely to have data entry error with TAB values. However, some systems may not be able to download data with this delimiter.

# Approach II: Combine Account Balance Data from Separate Files



- Sometimes specific policy information, such as account balances for investment-linked products, are available in a separate file. To include this information in the MPF, we can classify the file as “Additional Records”.
- We need to specify the input format of the main data file (which this additional data file is attached to) under “Record Format To Attach”. Please note that we may require longer run time especially the additional data file is large.
- If the policy number is used as the matching field, we need to select this field under “Cross Reference Variables” so that DCS knows how to combine these two data files. However, we are unable to specify a combination of multiple fields as the matching variables.

# Approach III: Combine Basic & Rider as One Record



- To combine (merge) two data files from basic & riders, we need to define a “Special Main Record to enable merging of files”, which is without any field. Both data files are defined as “**Sub Record**”.
- Under this “Main” input format, we need to specify a common field from both data files that can be used for merging (such as policy number). Similarly, we can only specify ONE field for data merging.
- We need to specify “Max Number of Sub Record” to allow for multiple records with the same policy numbers. Use `SUPPRESS_RECORD` if necessary, in order to overcome duplications in MPF.
- Fields from each data file are referred as *FieldName[x]* in the main codes, which *x* can be defined using `FOR ... ENDFOR` statements (e.g. `FOR x FROM 1 TO NO_OF_RECORDS("InputFormat")`). Please note that these fields should not be treated as array variables. For array variables, the variable index need to be defined as **constants**.

# Disclaimer

- The contents of this document are only for the reference of participants attending ASM Mini Sharing on 09.04.2019. Unauthorized circulation or duplication of any part of this document to any unauthorized third party is prohibited.
- The copyrights of this document are owned by ACTBuilder Actuarial Solutions Sdn. Bhd.
- ACTBuilder Actuarial Solutions Sdn. Bhd is not responsible for unauthorized use of any part of the contents of this document.

## Contact Info:

ACTBuilder Actuarial Solutions Sdn. Bhd. (1108112-P) | Lim Chee Beng | [CheeBeng.Lim@actbuilder.com.my](mailto:CheeBeng.Lim@actbuilder.com.my)