**Folder Structure**

$(Mold EX-Press for Mold SDK)

Bin\vs9 (vs10, vs11, vs14)

redist: vc runtime library

Custom\_info\_GML.csv: the file used to define custom properties

lang\_xxx.ini: language files

setting.ini: setting file

[Sample for Mold].exe

Include\COut.h: Mold EX-Press for Mold API header file

Source: [Sample for Mold] source code

Help\HELP\_EN.chm, 帮助文档\_CHS.chm

**[Sample for Mold] User Guide**

[Sample for Mold].exe of different MSVC versions can be found in $(Mold EX-Press for Mold SDK)\Bin\vs9 (vs10, vs11, vs14); if user is using an unsupported version of MSVC, the source code of sample can be found in $(Mold EX-Press for Mold SDK)\Source; user can directly upgrade the vs9 project to their MSVC version and build.

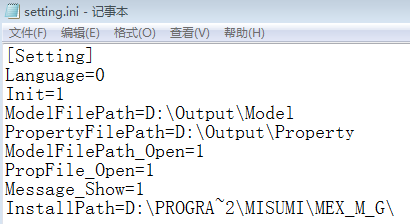
1. **Prerequisite**

Install Mold EX-Press for Mold (Global);

After installation, Mold EX-Press for Mold (Global) need to be run once in order to register Mold EX-Press for Mold API.

1. **Configuration File**

setting.ini is located in the same folder as [Sample for Mold].exe; it is used to save the current configuration of [Sample for Mold].



Language: the current value of “Language Setting” (seeing 3. Language Setting);

Init: the current initialization value of Mold EX-Press for Mold API (seeing 5. Interfaces initialization);

ModelFilePath, PropertyFilePath, ModelFilePath\_Open, PropFile\_Open: the current value of “File Path” (seeing 4. File Path Setting);

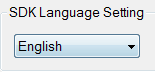
Message\_Show: used to control if popping the file path message dialog (seeing 6.A.4.a);

InstallPath: log the install path of Mold EX-Press for Mold (Global); when initializing, [Sample for Mold] will query the install path of Mold EX-Press for Mold (Global) in the registry, and log the path in Setting\InstallPath variable in setting.ini; user can refer to the codes of [Sample for Mold] initialization to get the install path of Mold EX-Press for Mold (Global).

1. **Language Setting**

[Sample for Mold] supports 6 languages by now, which are Simplified Chinese, English, Japanese, Traditional Chinese, Korean, Thai.

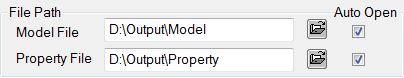
The current language setting is logged in Setting\Language variable in setting.ini.



1. **File Path Setting**

User can set the saving path for the generated model files and property files in “File Path”; if “Auto Open” is checked, [Sample for Mold] will open the saving folder automatically after files are generated.

The current “File Path” setting is logged in Setting\ModelFilePath, PropertyFilePath, ModelFilePath\_Open, PropFile\_Open in setting.ini;



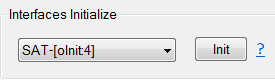
1. **Interfaces initialization**

User can select different initialization parameters in combo box under “Interfaces Initialize”; different initialization parameters will decide what formats of model files Mold EX-Press for Mold API will generate.

|  |  |
| --- | --- |
| Initialization parameters | Generated formats |
| SAT&DWG-[oInit:1] | sat, dwg |
| SAT-[oInit:4] | sat |
| SAT&STP-[oInit:5] | sat, stp |
| SAT&DWG&X\_T-[oInit:6] | sat, dwg, x\_t |

For instance, in order to generate only sat format user can select “SAT-[oInit:4]”, then click “Init” button; [Sample for Mold] will pop message box to notify the user if Mold EX-Press for Mold API is initialized successfully.

The current initialization parameter is logged in Setting\Init variable in setting.ini.



Note: Mold EX-Press for Mold API can only be initialized once in one process; so if initialization is successful, the “Init” button will become inactive.

1. **Import & Setting**

User select parameters in main combo box and sub combo box under “Import & Setting”, then click “Run” button; [Sample for Mold] will pop appropriate interface of Mold EX-Press for Mold.

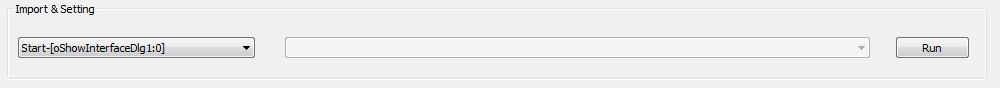
The corresponding relationship of the parameters in main\sub combo box with the interfaces of Mold EX-Press for Mold is shown in following table:



1. **Import:**

Import functions include “Start”, “Part Group 1”, “Part Group 2”, “Part Group 3”.

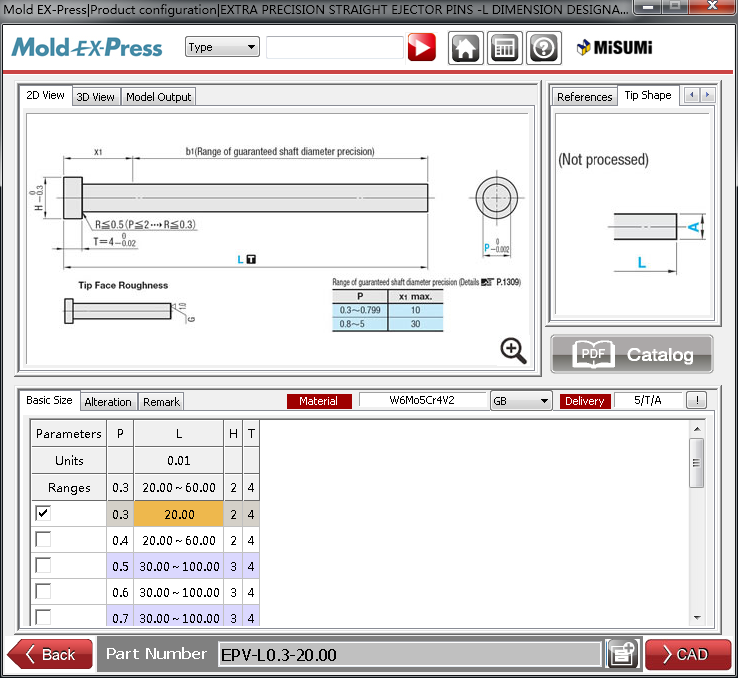
Take “Start” as an example to introduce the importing workflow:



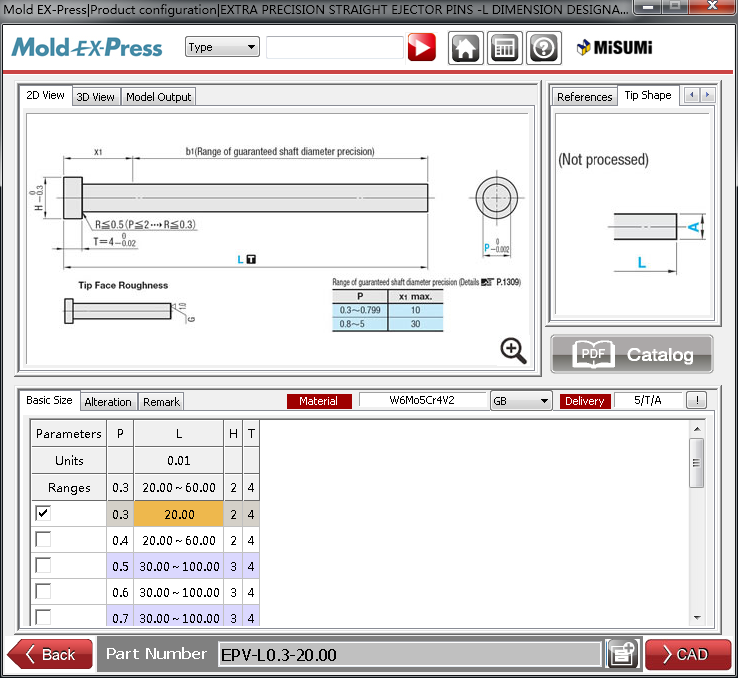
1. Select “Start-[oShowInterfaceDlg1:0]” in main combo box (sub combo box is unavailable in this situation), then click “Run” button; [Sample for Mold] will pop the first-class interface of Mold EX-Press for Mold.



1. Select “Straight ejector pins” category; click “Next” and go into the second-class interface of Mold EX-Press for Mold.

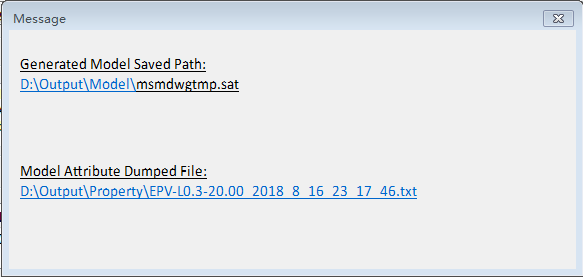


1. Select “Extra precision straight ejector pins”; click “Next” and go into the third-class interface of Mold EX-Press for Mold.



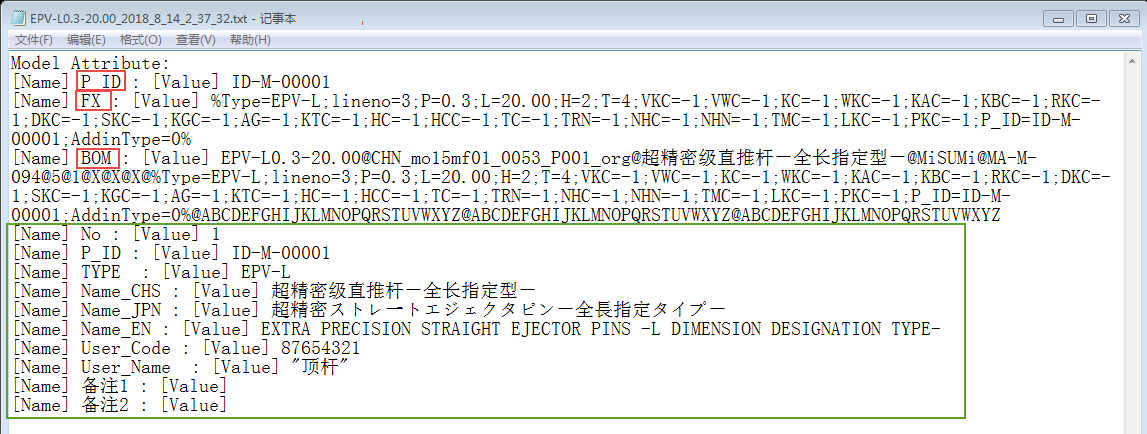
1. Click “CAD” in third-class interface; Mold EX-Press for Mold disappears and [Sample for Mold] will do following things automatically:
2. Pop message box to notify user the saving path for the model sat file and the txt file containing property information of the model;

The Setting\Message\_Show variable in setting.ini can be used to control if popping the file path message dialog.



sat file: model file generated by Mold EX-Press for Mold;

txt file: [Sample for Mold] extract property information of model by Mold EX-Press for Mold API and dump into txt file. The content of the txt file is as following:



The string following [Name] is the property name and that following [Value] is the property value; one-one correlation exists between the property name and value.

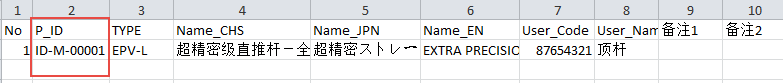
“P\_ID” means the ID of model, which can be used to search the custom properties for the model in Custom\_info\_GML.csv;

“FX” means the “modify string” of the model;

“BOM” means the “BOM string” of the model.

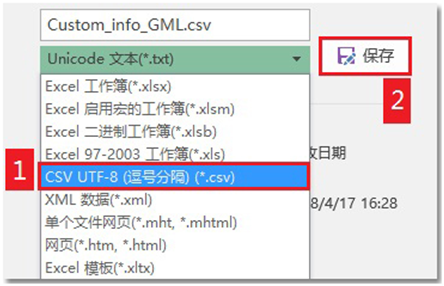
1. Custom Properties;

The name-value pairs below “BOM” are custom properties of the model; user can add custom properties for the model in Custom\_info\_GML.csv;

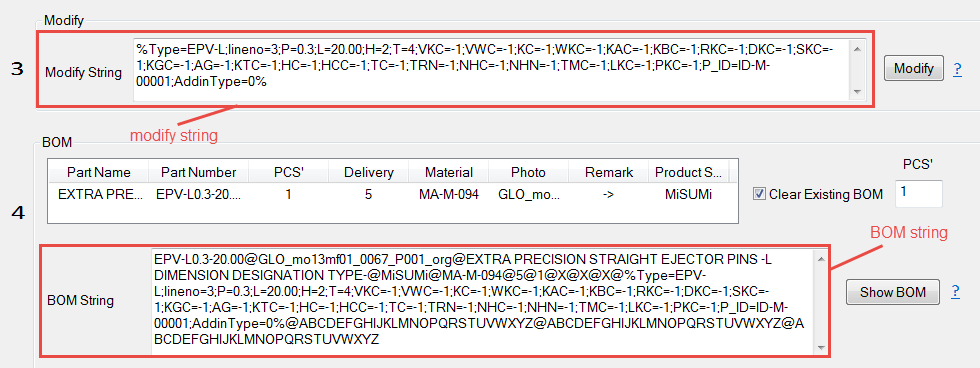


[Sample for Mold] will parse the custom properties in Custom\_info\_GML.csv during initialization; if one model is imported, [Sample for Mold] will search the custom properties for the model by P\_ID, then output these properties into the property txt file.

Notice: regardless of which editor the user is using to open Custom\_info\_GML.csv, after editing the csv file has to be saved as UTF-8 format, or else the Unicode characters may be unsupported. The following picture show the saving setting if editing Custom\_info\_GML.csv with MS Excel:



1. Generate “modify string” of the model and fill in “Modify String” edit box;
2. Generate “BOM string” of the model and fill in “BOM String” edit box.

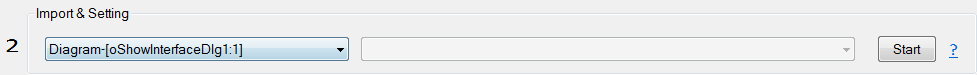


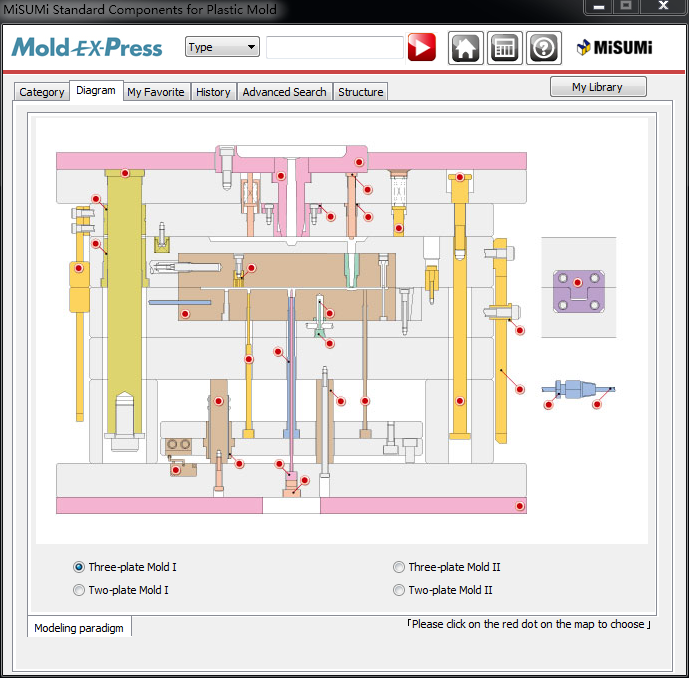
“Part Group 1”, “Part Group 2” and “Part Group 3” will import models as the same workflow; the only difference is Part Group 1\2\3 and their sub combo box will allow user to go directly into the second-class interface for each category.

1. **Utility:**

Utility includes “Diagram”, “My Favorite”, “History”, “Advanced Search”, “Part Name Directory”. Take “Diagram” as an example to introduce how to start the corresponding interface of Mold EX-Press for Mold.

1. Select “Diagram-[oShowInterfaceDlg1:1]” in main combo box (sub combo box is unimplemented in this situation), then click “Run” button; [Sample for Mold] will pop “Diagram” interface of Mold EX-Press for Mold.





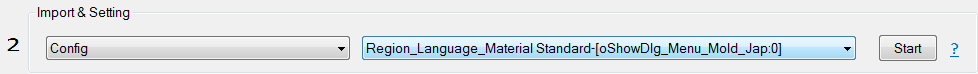
“My Favorite”, “History”, “Advanced Search” and “Part Name Directory” will start corresponding interface of Mold EX-Press for Mold with the same workflow.

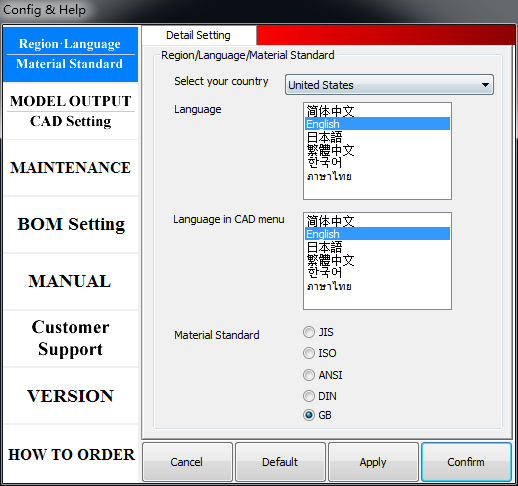
1. **Setting:**

Setting includes “Config”, “Help”.

Take language setting as an example to introduce how to start the corresponding interface of Mold EX-Press for Mold.

1. Select “Config” in main combo box; select “Region\_Language\_Material Standard- [oShowDlg\_Menu\_Mold\_Jap:0]” in sub combo box; then click “Run” button; [Sample for Mold] will pop “Region\_Language\_Material Standard” setting interface of Mold EX-Press for Mold.



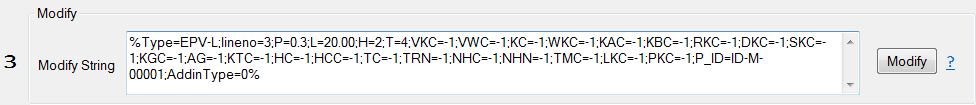


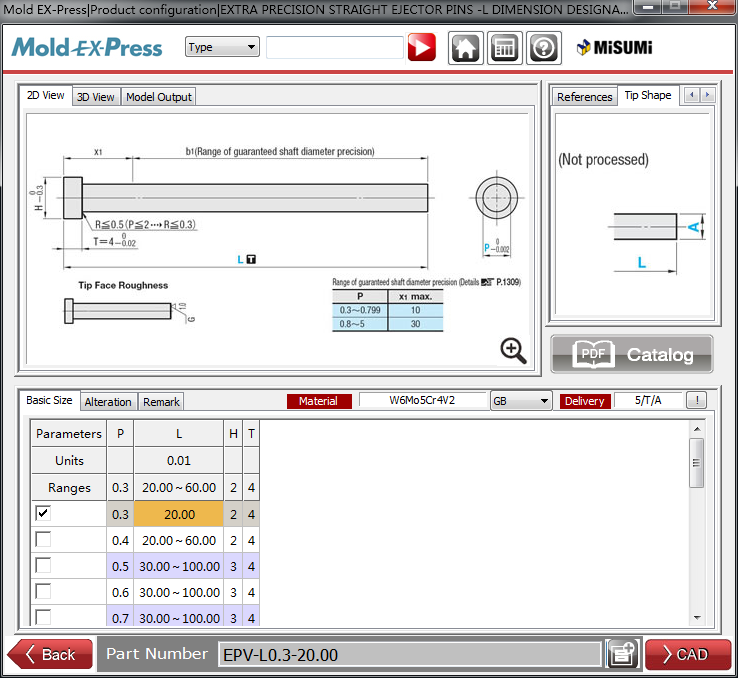
Other settings will start the corresponding interface of Mold EX-Press for Mold with the same workflow.

1. **Modify**

User input “modify string” for the specific model in “Modify String” edit box, then click “Modify” button; [Sample for Mold] will pop the third-class interface of Mold EX-Press for Mold for this model.

Take the imported “Straight ejector pins” -> “Extra precision straight ejector pins” above as an example, the corresponding third-class interface of Mold EX-Press for Mold is as following:

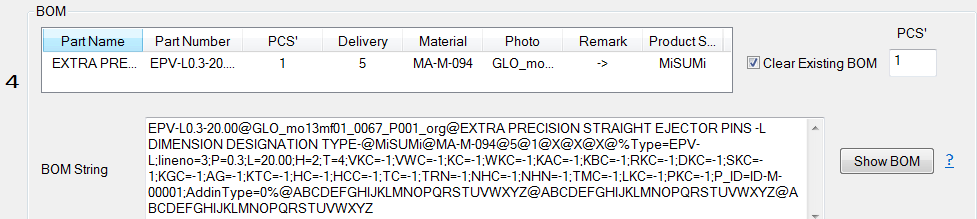




User can switch model size in this third class interface; click “CAD” to regenerate new model.

1. **BOM**

User first input “BOM string” for the specific model in “BOM String” edit box, then input part number in “Number” edit box (part number must be an integer greater than 0), finally click “BOM” button; [Sample for Mold] will add this model into BOM of Mold EX-Press for Mold.



Take the imported “Straight ejector pins” -> “Extra precision straight ejector pins” -> “EPV-L0.3-20.00” above as an example, the default “BOM String” is:

EPV-L0.3-20.00@GLO\_mo13mf01\_0067\_P001\_org@EXTRA PRECISION STRAIGHT EJECTOR PINS -L DIMENSION DESIGNATION TYPE-@MiSUMi@MA-M-094@5/T/A@1@X@X@X@%Type=EPV-L;lineno=3;P=0.3;L=20.00;H=2;T=4;VKC=-1;VWC=-1;KC=-1;WKC=-1;KAC=-1;KBC=-1;RKC=-1;DKC=-1;SKC=-1;KGC=-1;AG=-1;KTC=-1;HC=-1;HCC=-1;TC=-1;TRN=-1;NHC=-1;NHN=-1;TMC=-1;LKC=-1;PKC=-1;AddinType=0%@ABCDEFGHIJKLMNOPQRSTUVWXYZ@ABCDEFGHIJKLMNOPQRSTUVWXYZ@ABCDEFGHIJKLMNOPQRSTUVWXYZ

1 in red means the number of part is 1. If user input 5 in “Number” edit box, 1 in red in “BOM string” will be replaced by 5.

“Clear Existing BOM” check box allow user to control if to clear all existing contents in BOM before adding this model.

The BOM for “Straight ejector pins” -> “Extra precision straight ejector pins” -> “EPV-L0.3-20.00” with “Number” 5 and “Clear Existing BOM” checked will be:

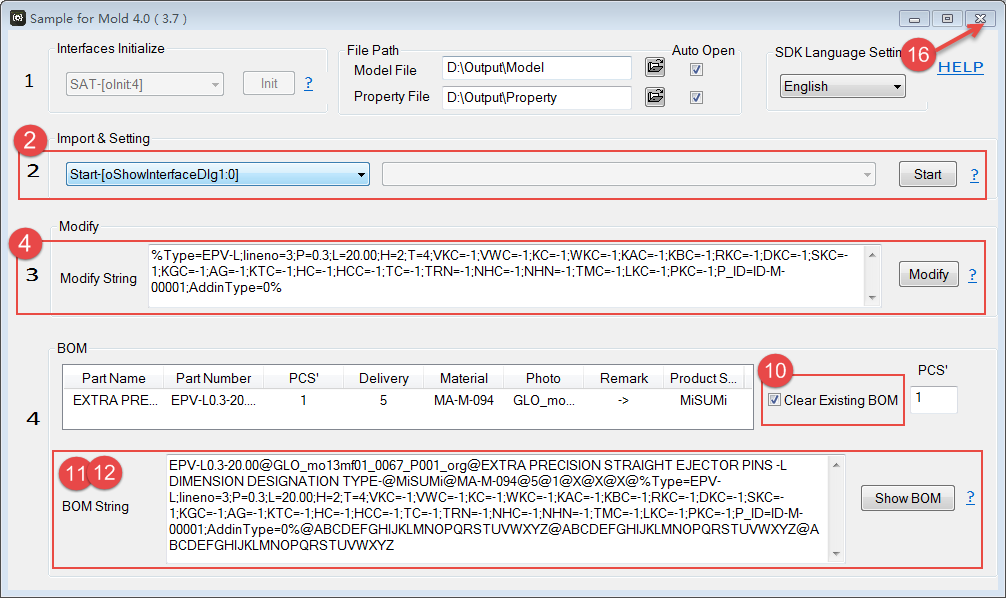


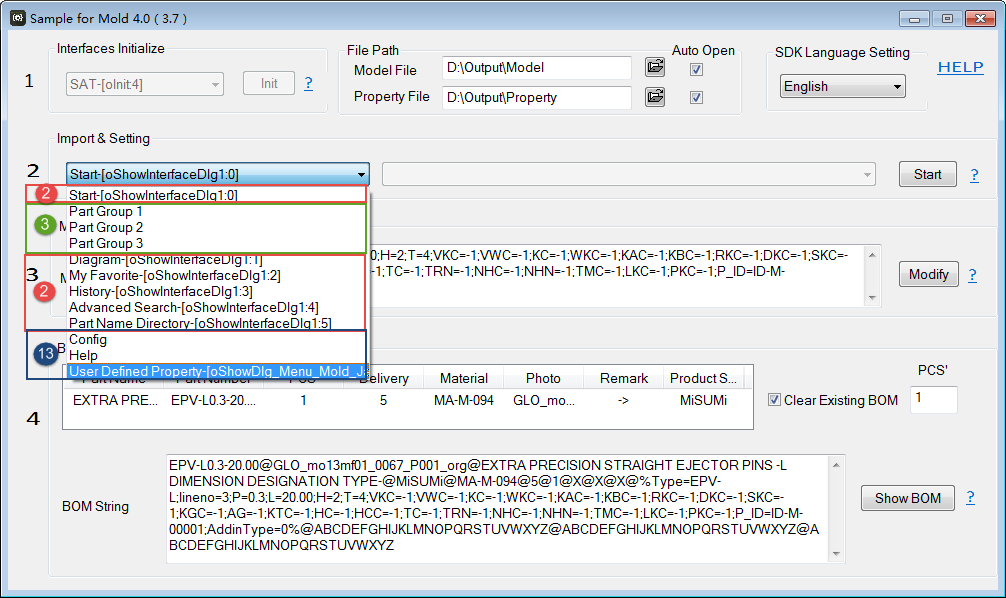
**Mold EX-Press for Mold API**

All exported API functions are contained in COut.h;

User can find COut.h in $(Mold EX-Press for Mold SDK)\include.

Following two pictures show the mapping between APIs and the functionalities in [Sample for Mold].





1. void oInit(short i)

Role: initialize Mold EX-Press for Mold API; can only be call once in one process.

Input Parameters:

i: value range is parameters in following table; different parameters can decide what formats of model files Mold EX-Press API will generate.

|  |  |
| --- | --- |
| Parameters | File formats |
| 1 | sat, dwg |
| 4 | sat |
| 5 | sat, stp |
| 6 | sat, dwg, x\_t |

1. short oShowInterfaceDlg1(short i)

Role: call the first-class interface of Mold EX-Press for Mold.

Input Parameters:

i: value range is integer from 0 to 5; Mold EX-Press for Mold will call the corresponding first-class interface according to the parameters.



Return Values:

short: meaningless; Mold EX-Press for Mold use it internally.

1. short oShowInterfaceDlg2(short index, BOOL C\_Value, short Setting)

Role: call the second-class interface of Mold EX-Press for Mold for each category.

Input Parameters:

index: value range is integer from 0 to 34; Mold EX-Press for Mold call the second-class

interface for corresponding category according to the parameters.



C\_Value: FALSE; fixed value.

Setting: 0; fixed value.

Return Values:

short: meaningless; Mold EX-Press for Mold use it internally.

1. short oShowInterfaceDlg3(LPCTSTR strPara)

Role: call the third-class interface of Mold EX-Press for Mold for specific model; used for modifying.

Input Parameters:

strPara: the modify string for specific model

Return Values:

short: meaningless; Mold EX-Press for Mold use it internally.

1. short oGetParaCount()

Role: get the number of properties in model.

Return Values:

short: the number of properties in model

1. CString oGetParaName(short index)

Role: get the property name at index position in model.

Input Parameters:

index: value range is integer from 0 to the number of properties; an one-one correlation exist between the property name and value at the same index position.

Return Values:

CString: the property name at index position in model

1. CString oGetParaVlaue(short index)

Role: get the property value at index position in model.

Input Parameters:

index: value range is integer from 0 to the number of properties; an one-one correlation exist between the property name and value at the same index position

Return Values:

CString: the property value at index position in model

1. CString oGetTypeName()

Role: get the Type of model.

Return Values:

CString: the Type of model

Note: the return Type can be used to check if end-user import model successfully; if the Type is null, that means end-user close Mold EX-Press for Mold before importing.

1. CString oGetOrder()

Role: get the Order of model.

Return Values:

CString: the Order of model

1. void oClearBom()

Role: clear all existing contents in BOM.

1. void oAddToBom(LPCTSTR cPara)

Role: add the specific model into BOM.

Input Parameters:

cPara: the BOM string for the specific model

1. void oShowBom()

Role: call BOM interface of Mold EX-Press for Mold.

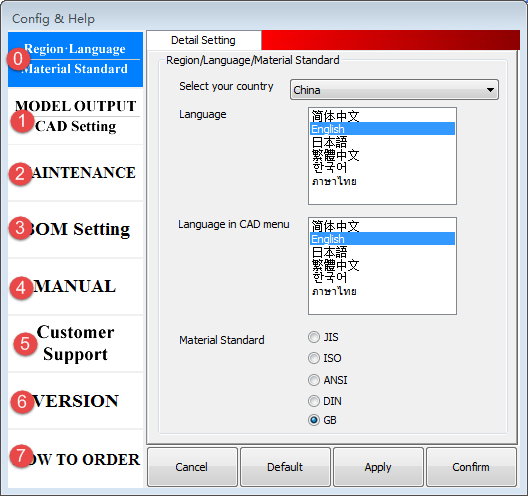
1. void oShowDlg\_Menu\_Mold\_Jap(short VALUE)

Role: call setting interfaces of Mold EX-Press for Mold.

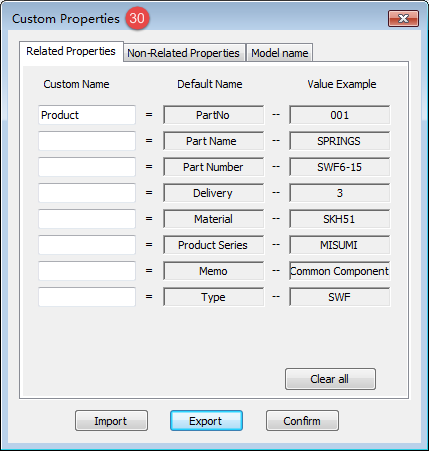
Input Parameters:

VALUE: value range is integer from 0 to 7, 30; Mold EX-Press for Mold will call the corresponding interfaces according to the parameter.

If VALUE is 0 to 7, Mold EX-Press for Mold will call the corresponding setting interfaces;



If VALUE = 30, Mold EX-Press for Mold will call the “Customer Properties” interface;



1. void oShowProductAddDlg()

Role: call “Add Customized BOM info” interface of Mold EX-Press for Mold.

1. short oGetExeState()

Role: get the initialization parameter used for oInit in current process.

Return Values:

short: value range is integer from 0 to 6; refer to oInit for the meaning of the value.

1. void oUninit()

Role: terminate Mold EX-Press for Mold API; Can only be call once in one process.