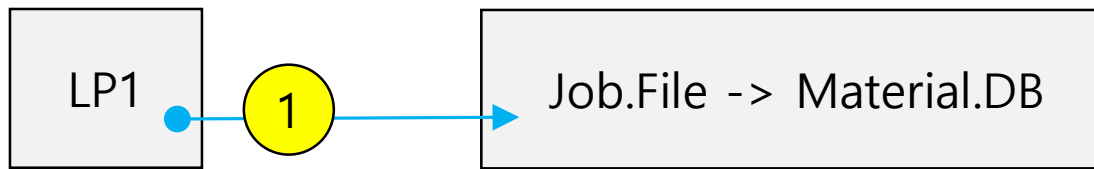


Scheduler 관련 주요 API 설명

관련 Class SCX__CTC__SCH_MATERIAL_CTRL
SCI__CTC__SCH_MATERIAL_CTRL*

Scheduler내 Wafer 기본 정보 :

- Position
- Status



1

SCI__CTC__JOB_FILE_CTRL

* Job 파일을 관리하는 Class 입니다.

int Link__Material_DB(const CString& service_name);

* Service Name을 통해 생성된 Material DB에 연결한다.

* Material DB의 Service Name은 아래의 class 함수를 통해 생성된다.

SCI__CTC__SCH_MATERIAL_CTRL

int Create__SERVICE_DB(const CString& service_name);

int Upload__JOB_FILE(const CString& dir, const CString& port, const CString& file);

* .job 파일

<- *.css 파일

<- *.rte 파일

* job 관련 모든 정보를 upload 한다.

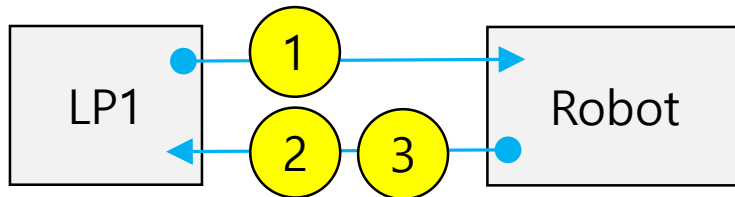
void Copy__TO_MATERIAL(const CString& job_name);

* 위의 함수를 통해 job 관련 업로드된 정보들을 연결된 Material DB에 복사한다.

* 내부적으로 아래의 class 함수를 통해 이루어진다.

CRes__SCH_MATERIAL_INFO

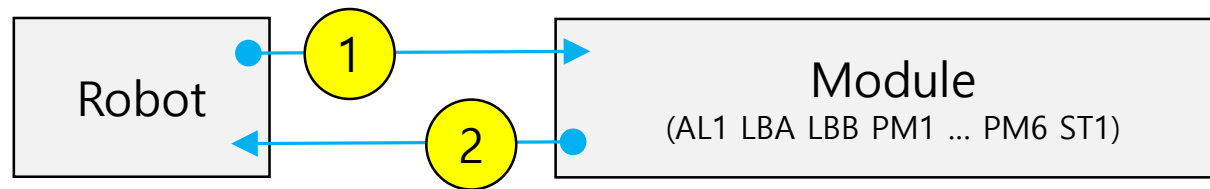
int DB_CTRL__Create_MATERIAL(...);



- 1 `int Pick_From_LPx(const int port_id, const int slot_id, const CString& robot_pos);`
- `port_id` : load port의 id를 의미함. (Range : 1 ~ 5)
 - `slot_id` : load port내 Wafer가 있는 slot numbe를 의미함. (Range : 1 ~ 25)
 - `robot_pos` : robot이 load port에서 어떤 arm으로 pick했는지 의미함. (Range : A B)

- 2 `int Place_To_LPx(const CString& robot_pos, const CString& module_pos);`
- `robot_pos` : robot이 load port에 어떤 arm으로 place하는지 의미함. (Range : A B)
 - `module_pos` : wafer가 place 될 load port module을 의미함.
- Format) LPx-Slot
- Ex.) LP1-1 ~ LP1-25

- 3 `int Clear_MATERIAL_INFO(const int port_id, const int slot_id);`
- 해당 Wafer 정보가 Scheduler DB에서 제거된다.



1

int Place_To_MODULE(const CString& robot_pos, const CString& module_pos);

- robot_pos : robot이 특정 module에 wafer를 place할 때 사용하는 arm. (Range : A B)
- module_pos : wafer가 투입될 module 을 의미함.

Format.) module-slot

Ex.) AL1-1, LBA-2

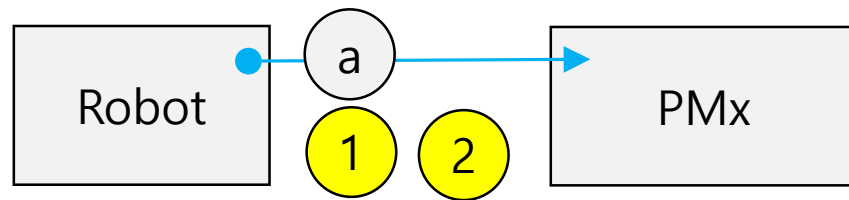
2

int Plick_From_MODULE(const CString& module_pos, const CString& robot_pos);

- module_pos : wafer를 pick할 module 을 의미함.

Ex.) ST1-11, LBB-1

- robot_pos : robot이 특정 modul에서 wafer를 pick할 때 사용하는 arm. (Range : A B)



a

Robot이 Wafer를 PM에 투입한 후에 아래의 동작을 수행합니다.

1

int Goto__NEXT_PROCESS(const CString& material_pos);

- material_pos : wafer가 투입된 process module.

Ex.) PM1-1

- wafer에서 수행될 다음 recipe 정보로 이동한다.

2

int Set__MATERIAL_STATUS(const CString& material_pos, const CString& status);

- status : wafer의 상태 값을 설정한다.

* wafer를 process module에 place 했을 때 : PROCESSING

* wafer를 process module에서 pick 했을 때 : PROCESSED

```
int Get__MATERIAL_STATUS(const CString& material_pos, IDS__SCH_MATERIAL_STATUS& ds_info);
```

- . 해당 Material 위치의 아래 정보를 받아온다.

```
class IDS__SCH_MATERIAL_STATUS
```

```
int iPORT_ID; // wafer의 port id
```

```
int iSLOT_ID; // wafer의 slot id
```

```
CString sMATERIAL_POS; // 현재 Wafer가 있는 위치 정보
```

```
int Get__MATERIAL_INFO(const CString& material_pos, IDS__SCH_MATERIAL_INFO& ds_info);
```

- . 해당 Material 위치의 아래 정보를 받아온다.

```
class IDS__SCH_MATERIAL_INFO
```

```
int iSRC_PTN;    // wafer의 source port id
```

```
int iSRC_STN;    // wafer의 source slot id
```

```
int iTRG_PTN;    // wafer의 target port id
```

```
int iTRG_STN;    // wafer의 target slot id
```

```
CString sCUR_RECIPE;    // wafer에서 수행할 process recipe
```

```
CString sLOT_ID;        // wafer에 부여된 LotID
```

```
Cstring sMATERIAL_ID;   // wafer에 부여된 MaterialID
```

```
CString sPPID;          // wafer에 부여된 PPID (route recipe)
```

```
CString sMATERIAL_TYPE; // water에 부여된 material type
```

```
int Get_CUR_PROCESS_INFO(const CString& material_pos, CStringArray& l_prc_module, CStringArray& l_prc_recipe);
```

- 해당 Material 위치의 process 관련 정보를 받아온다.

- l_prc_module : 공정 진행할 process module list

 - Ex.) PM1 ~ PM6

- l_prc_recipe : 공정 진행할 process recipe list