**Functional Specification – SAP MII Phase 1**

Project: Replace Mill Production Schedule, Slab Reapplication, Delays and Coil Rework Modules

|  |  |
| --- | --- |
| ***Phase:*** | Business Blueprint |
| ***Status:*** | Private |
| ***Author:*** | Luciano Gonzalez, Crispin Ponce |
| ***Date:*** | May/08/2017 |
| ***Copyright:*** | © Neoris |
| ***Company:*** | NEORIS |
| ***Customer:*** | Northstar Bluescope Inc. |

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Description | Author |
| 1.0 | May/08/2017 | Document creation | Luciano Gonzalez  Crispin Ponce |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Index

[Functional Specification – SAP MII Phase 1 1](#_Toc486424717)

[1 Revision History 3](#_Toc486424718)

[2 Index 4](#_Toc486424719)

[1 Document Holds 8](#_Toc486424720)

[2 Introduction 9](#_Toc486424721)

[3 References 10](#_Toc486424722)

[4 General Introduction 11](#_Toc486424723)

[5 UC033 – Mill Production Scheduling - Display Available Schedules 12](#_Toc486424724)

[6 UC034 – Mill Production Scheduling - Display Available Schedules Detail 15](#_Toc486424725)

[7 UC035 – Mill Production Scheduling - Obtain Schedule 19](#_Toc486424726)

[8 UC036 – Mill Production Scheduling - Display Missed Production 28](#_Toc486424727)

[9 UC037 – Mill Production Scheduling - Display Order Book 31](#_Toc486424728)

[10 UC038 – Mill Production Scheduling - Display Requires Disposition 35](#_Toc486424729)

[11 UC039 – Mill Production Scheduling - Display Stock 39](#_Toc486424730)

[12 UC040 – Mill Production Scheduling - Display Casters 42](#_Toc486424731)

[13 UC041 – Mill Production Scheduling - Display Hot Strip Mill Schedule 47](#_Toc486424732)

[14 UC042 - Mill Production Scheduling - Display Caster Graph 52](#_Toc486424733)

[15 UC043 - Mill Production Scheduling - Display Hot Strip Mill Graph 54](#_Toc486424734)

[16 UC044 - Mill Production Scheduling - Display Workspace 56](#_Toc486424735)

[17 UC045 - Mill Production Scheduling – Save 58](#_Toc486424736)

[18 UC046 - Mill Production Scheduling - Print 63](#_Toc486424737)

[19 UC047 - Mill Production Scheduling - Refresh 66](#_Toc486424738)

[20 UC048 - Mill Production Scheduling - Heat List 68](#_Toc486424739)

[21 UC049 - Mill Production Scheduling - PDI Update / Retransmit PDI 72](#_Toc486424740)

[22 UC050 - Mill Production Scheduling - Width Verification – Not Used 74](#_Toc486424741)

[23 UC051 - Mill Production Scheduling -Transition Slab – Not Used 75](#_Toc486424742)

[24 UC052 - Mill Production Scheduling -Delays – Not Used 76](#_Toc486424743)

[25 UC053 - Mill Production Scheduling – Display Alternate Grade 77](#_Toc486424744)

[26 UC054 - Mill Production Scheduling - Select Alternate Grade 79](#_Toc486424745)

[27 UC055 - Mill Production Scheduling - Check Schedule – Not Used 83](#_Toc486424746)

[28 UC056 - Mill Production Scheduling - Release Schedule 84](#_Toc486424747)

[29 UC057 - Mill Production Scheduling - AutoRelease Schedule 89](#_Toc486424748)

[30 UC058 - Mill Production Scheduling - Release Slab – Not Used 91](#_Toc486424749)

[31 UC059 - Mill Production Scheduling - Suspend Interfaces 92](#_Toc486424750)

[32 UC060 - Mill Production Scheduling - Add/ Remove pieces to Schedule – Not Used 94](#_Toc486424751)

[33 UC061 - Mill Production Scheduling - Display Piece Detail 95](#_Toc486424752)

[34 UC062 – Slab Re-Application – Display Slabs in the Tunnel Furnace 104](#_Toc486424753)

[35 UC063 – Slab Re-Application – Display Matching Committed Coils for a Slab 106](#_Toc486424754)

[36 UC064 – Slab Re-Application – Display Matching Required Disposition Coils for a Slab (Implemented on UC063) 114](#_Toc486424755)

[37 UC065 – Slab Re-Application – Display Matching Stock Coils for a Slab (Implemented on UC063) 115](#_Toc486424756)

[38 UC066 – Slab Re-Application – Re-Apply Slab 116](#_Toc486424757)

[39 UC067 – Slab Re-Application – Change search criteria for matching orders 130](#_Toc486424758)

[40 UC068 – Slab Re-Application – Display Slab Details 133](#_Toc486424759)

[41 UC069 – Slab Re-Application – Display Order Details 138](#_Toc486424760)

[42 UC070 – Slab Re-Application – Retransmit PDI to rolling mill 141](#_Toc486424761)

[43 UC071 - Slab Re-Application – Undo a slab re-application 143](#_Toc486424762)

[44 UC072 - Slab Re-Application - Resend Schedule 152](#_Toc486424763)

[45 UC073 – Coil Rework - Refresh Yard 154](#_Toc486424764)

[46 UC074 - Coil Rework – Input Mandatory Checks 156](#_Toc486424765)

[47 UC075 - Coil Rework – Send 159](#_Toc486424766)

[48 UC076 - Delays – Record Delay (manually) 163](#_Toc486424767)

[49 UC077 - Delays – Record Delay (automatic) 168](#_Toc486424768)

[50 UC078 - Delays – Assign a Delay Reason 173](#_Toc486424769)

[51 UC079 - Delays – Split a Delay Event 176](#_Toc486424770)

[52 UC080 - Delays – Record Slowdown (automatic) 179](#_Toc486424771)

[53 UC081 - Delays – Assign a Slowdown Reason 184](#_Toc486424772)

[54 UC082 - Delays – Split a Slowdown Event 187](#_Toc486424773)

[55 UC083 – Master Data – Delay Reasons Maintenance 190](#_Toc486424774)

[56 UC084 – Master Data – Slowdown Reasons Maintenance 194](#_Toc486424775)

[57 UC085 – Master Data – Areas Maintenance 198](#_Toc486424776)

[58 UC086 – Master Data – Disciplines Maintenance 201](#_Toc486424777)

[59 UC087 – Master Data – Equipment Maintenance 205](#_Toc486424778)

[60 UC088 – Delays – Calculate Crew 208](#_Toc486424779)

[61 UC089 – Delays – Slowdown and Delay Report 211](#_Toc486424780)

[62 UC090 – Mill Production Scheduling – Drag and Drop into Casters List 213](#_Toc486424781)

[63 UC091 – Mill Production Scheduling – Drag and Drop into Missed Production 216](#_Toc486424782)

[64 UC092 – Mill Production Scheduling – Drag and Drop into Stock 219](#_Toc486424783)

[65 UC093 – Mill Production Scheduling – Drag and Drop into Required Disposition 222](#_Toc486424784)

[66 UC094 – Mill Production Scheduling – Drag and Drop into Workspace 225](#_Toc486424785)

[67 UC095 – Mill Production Scheduling – Drag and Drop into OrderBook 228](#_Toc486424786)

1. Document Holds

|  |  |  |  |
| --- | --- | --- | --- |
| Hold No. | Section No. | Description | Status |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Introduction
   1. **Purpose**

The purpose of this document is to describe the functionality to be developed in MII in order to replace the current applications that North Star BlueScope is using to control their production. The applications are:

* Automatic Data Collection Applications
* Mill production Scheduling
* Slab re-application
* Coil re-work
* Delay application
* BEA message queue system.
  1. **Scope**

This document covers the functionality requested on RFP for phase 1.

* 1. **Definitions, Acronyms y Abbreviations**

|  |  |  |
| --- | --- | --- |
| Definition | Description | Status |
| Ladle |  |  |
| EAF |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. References
   1. **Factory Database Entity Relationship Diagram**

Please refer to the SAP MII Phase 1 – RFP (Final).docx document.

* 1. **MII Security, Roles, and Custom Attributes**

Please refer to the Roles\_Users\_Attributes\_Setup-NSBS.xlsx document.

1. General Introduction

## Introduction

This section of the specification coverts the interfaces needed to communicate the MII System with level 2 applications and with Level 3 applications.

## MII Communication

The MII application communicates with level 2 systems through a message broker called PCo (Plant Connectivity). It creates connections with each of the different systems, supporting several communication protocols like text files, OPC, web services .

# UC033 – Mill Production Scheduling - Display Available Schedules

## Description

This functionality shows the available schedules to the user, so he can select one of the schedules to get the detailed information.

## Triggers

* When the user clicks on the “obtain schedule” button.

## Preconditions

* User must be on the “Production Schedule” dashboard.

## Actors

* User
* Scheduling System (SS001)

## Basic Flow

## Basic Flow

## The actor enters to the Production Schedule screen.

## The actor clicks on the “Obtain Schedule” button

## The system obtains the available schedules from the Scheduling System (BR1) and opens a new window displaying the list of Available Schedules with the following information:

* Schedule Number
* Schedule Description

## Use case ends.

## Alternative and Exception Flows

## There are no alternate flows

## Business Rules

## BR1 – Schedules available to MPS

## Only the schedules that have an “S” status, are shown on the list

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen**  **/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Fields** |
| Schedule Number | Output | Yes | N/A | None | SB\_HSM\_SCHEDULE | HSM\_SCHEDULE\_NO |
| Description | Output | No | N/A | None | SB\_HSM\_SCHEDULE | HSM\_SCHEDULE\_DESCRIPTION |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## 

## Flow Chart

None.

## Technical Specification

## To get the data for the schedules use:

## SELECT hsm\_schedule\_no, hsm\_schedule\_description FROM sb\_hsm\_schedule WHERE prod\_status = "S" ORDER BY prod\_seq.

# UC034 – Mill Production Scheduling - Display Available Schedules Detail

## Description

This functionality shows the detailed information of the schedule selected from the schedule list.

## Triggers

* When the user clicks on the “Obtain” button.

## Preconditions

* User must use UC033.

## Actors

* User
* Scheduling System (SS001)

## Basic Flow

## Basic Flow

## The actor selects a schedule from the list.

## The actor clicks on the “Obtain” button

## The system obtains the schedule detail from the selected schedule and validates it (BR1) (AF1).

## The system Displays the Schedule Detail from the selected schedule with the following information:

* Schedule Number
* Schedule Description
* Estimated Production Time
* Starting Width
* Starting Grade
* Number of heats

## Use case ends.

## Alternative and Exception Flows

## AF1 – Validation Fails.

## The system displays an error message

## The user clicks on “OK” button

## Error message is closed.

## End of use case.

## AF2 - User clicks on the “Cancel” button.

## The “Schedule Detail” modal window is closed

## End of use case.

## Business Rules

## BR1 – Schedule Validation

## When obtaining the schedule, the system must validate the following:

## Quantity of slabs with status “normal” obtained for that schedule is greater than 0 and has the same number of coils in the schedule

## Quantity of heats assigned to the schedule is greater than 0, and is the same number as the schedule selected

## Quantity of the caster string is greater than 0 and is the same number as the schedule selected.

## If any of these validations fails, the Schedule must be updated to status “A” (Available). No schedule is loaded nor displayed on the screen.

## If all validations where correct, the Schedule must be updated to status “M” (Approved), set the last slab from the schedule as “last” before tundish change.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Schedule Number | Output | Yes | N/A | N/A | SB\_HSM\_SCHEDULE | HSM\_SCHEDULE\_NO |
| Description | Output | No | N/A | N/A | SB\_HSM\_SCHEDULE | HSM\_SCHEDULE\_DESCRIPTION |
| Estimated Production Time | Output | Yes | N/A | N/A | SB\_HSM\_SCHEDULE | EST\_PROD\_TIME |
| Starting Width | Output | Yes | N/A | 1 decimal place | CASTER\_STRING\_SLAB | SLAB\_WIDTH\_HEAD |
| Starting Grade | Output | Yes | N/A | N/A | HEAT\_DETAIL | GRADE |
| Number of Heats | Output | Yes | N/A | N/A | SB\_HSM\_SCHEDULE | NUMBER\_OF\_HEATS |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

The following queries are required to be executed (dlgavailableschedules.cpp line 113):

* Load the selected Schedule in memory:
  + SELECT hsm\_schedule\_no, hsm\_schedule\_description, approved\_dt\_stamp, approved\_user\_id, pattern\_id, est\_prod\_time, avg\_kmw, total\_coil\_tonnage, total\_coil\_length, number\_of\_strings, number\_of\_coils, schedule\_min\_coil\_weight, schedule\_max\_coil\_weight, schedule\_min\_width, schedule\_max\_width, schedule\_min\_gauge, schedule\_max\_gauge, strings, grades, avg\_casting\_time, transitions, heat\_tonnage, coil\_tonnage, prod\_seq, prod\_status, change\_datetime, change\_user\_id, slab\_width\_calculated, number\_of\_heats FROM sb\_hsm\_schedule WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]
* Validate the schedule
  + CasterStringSlabCount = SELECT COUNT(\*) FROM caster\_string\_slab where hsm\_schedule\_no=[SELECTED SCHEDULE] AND planned\_slab\_type=0
  + CasterStringCount = SELECT COUNT(\*) FROM caster\_string where hsm\_schedule\_no=[SELECTED SCHEDULE]
  + HeatDetailCount = SELECT COUNT(\*) FROM heat\_detail where hsm\_schedule\_no=[SELECTED SCHEDULE]
  + Validate if the CAsterStringSlabcount>0
    - Validate if the CasterStringCount>0
      * Then Validate if HeatDetailCount>0
        + Then Validate if CasterStringCount is different than the Number\_of\_strings Field from the previous query. If it is, it’s a fail and “Invalid Number of Strings”.
        + Else, validate if the HeatDetailCount is different to the number\_of\_heats field from the previous query. If it is, it’s a fail and “Invalid Number of Heats”
        + Else, validate if the CasterStringSlabCount is different to the number\_of\_coils field from the previous query. If it is, it’s a fail and “Invalid Number of coils”
      * If is not bigger than 0, then the Error, “Heat Detail is empty”, is returned.
    - If its not bigger than 0, then the error “Caster string is empty” is returned.
  + If its not bigger than 0, then the error “ Caster String Slab is empty” is returned.
  + If any of the validations above fails, then:
    - Begin SQL transaction
      * UPDATE sb\_hsm\_schedule SET prod\_status = 'A' change\_user\_id = [CURRENT USER change\_datetime = getdate() WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]
      * Validate if the HeatDetailCount>0. If it is:
        + UPDATE heat\_detail SET prod\_status = 'A' change\_user\_id = [CURRENT USER change\_datetime = getdate() WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]
        + Validate if the CasterStringCount>0. If it is:

UPDATE caster\_string SET prod\_status = 'A' change\_user\_id = [CURRENT USER change\_datetime = getdate() WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]

* + - * + Validate if the CasterStringSlabCount>0. If it is:

UPDATE caster\_string\_slab SET prod\_status = 'A' change\_user\_id = [CURRENT USER change\_datetime = getdate() WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]

* + - Commit the transaction
    - Refresh the schedules list:
      * SELECT hsm\_schedule\_no, hsm\_schedule\_description FROM sb\_hsm\_schedule WHERE prod\_status = 'S' RDER BY prod\_seq

# UC035 – Mill Production Scheduling - Obtain Schedule

## Description

This functionality allows to obtain all the information required to send the production schedule and Product Detail Input to the Level 2 as well as displaying it on the Production Schedule dashboard.

## Triggers

* When the user clicks on the “Confirm” button.

## Preconditions

* User must use UC034.

## Actors

* User
* Scheduling System (SS001)

## Basic Flow

## Basic Flow

## The actor clicks on the “Confirm” button (FA1) (BR1) (BR2)

## The system sends the Product Data Input messages to L2 (BR3).

## The system displays the Casters and Hot Strip Mill Schedule in two separated tabs, with the following information:

* Casters:
  + L3 Coil Id
  + HSM Sequence
  + Order
  + Slab Width Head
  + Slab Width Tail
  + Production Status
  + WC
  + TC
  + Date Orig
* Hot Strip Mill Schedule:
  + HSM Sequence
  + Order
  + Coil Thick
  + Coil Width
  + Coil Weight
  + Prod Stat
  + WC
  + TC
  + Casting Minutes
  + Date Orig

## Use case ends.

## Alternative and Exception Flows

## FA1 - User clicks on the “Cancel” button.

## The “Schedule Detail” modal window is closed

## End of use case.

## Business Rules

## BR1 – Schedule Validation

## When obtaining the schedule, the system must validate the following:

## Validate if the schedule is still sequenced, if not, show a warning to the user

## If the schedule is still sequenced, update the status on the planning tables to “M”.

## BR2 – Retrieve all Slab Data into current schedule data.

## The system must obtain all slabs for the schedule, and perform the following:

## Assign turnaround and roll change

## Re sequence the new schedule

## Build the heat list and save it

## Assign next L3 id for coils and heat

## Save all newly created slabs.

## BR3 – Messages to be Sent to L2.

## The following messages must be sent to L2:

## DEC\_L3\_ALL\_HEATS

## DEC\_L3\_ALL\_SLABS

## If there is no more coils added to the schedule DEC\_L3\_ALL\_COILS is sent, if there are more coils added then DEC\_MPS\_COIL\_ADDED will be sent.

## A transmission success or error message must be returned and logged.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Caster - L3 Coil Id | Output | Yes | N/A | N/A | caster\_string\_slab | Line\_item\_coil\_no |
| Caster - HSM Sequence | Output | Yes | N/A | N/A | caster\_string\_slab | Sched\_slab\_seq |
| Caster - Order | Output | Yes | N/A | N/A | caster\_string\_slab | Order\_no |
| Caster - Slab Width Head | Output | Yes | N/A | N/A | caster\_string\_slab | Slab\_width\_head |
| Caster - Slab Width Tail | Output | Yes | N/A | N/A | caster\_string\_slab | Slab\_width\_tail |
| Caster - Production Status | Output | Yes | N/A | N/A | caster\_string\_slab | Prod\_status |
| Caster - WC | Output | Yes | N/A | N/A | caster\_string\_slab | Slab\_width\_change\_mode |
| Caster - TC | Output | Yes | N/A | N/A | caster\_string\_slab | Precedes\_turn\_around |
| Caster – Date Orig | Output | Yes | N/A | N/A | Customer\_order\_line\_item | Promised\_date |
| Hot Strip Mill - HSM Sequence | Output | Yes | N/A | N/A | caster\_string\_slab | Hsm\_schedule\_no |
| Hot Strip Mill - Order | Output | Yes | N/A | N/A | caster\_string\_slab | Order\_no |
| Hot Strip Mill - Coil Thick | Output | Yes | N/A | N/A | caster\_string\_slab | Mill\_gauge\_aim |
| Hot Strip Mill - Coil Width | Output | Yes | N/A | N/A | caster\_string\_slab | Mill\_width\_aim |
| Hot Strip Mill - Coil Weight | Output | Yes | N/A | N/A | caster\_string\_slab | Mill\_weight\_aim |
| Hot Strip Mill - Prod Stat | Output | Yes | N/A | N/A | coil | Coil\_status |
| Hot Strip Mill - WC | Output | Yes | N/A | N/A | caster\_string\_slab | Slab\_width\_change\_mode |
| Hot Strip Mill - TC | Output | Yes | N/A | N/A | caster\_string\_slab | Precedes\_turn\_around |
| Hot Strip Mill - Casting Minutes | Output | Yes | N/A | N/A | caster\_string\_slab | Casting\_time |
| Hot Strip Mill - Date Origin | Output | Yes | N/A | N/A | Customer\_order\_line\_item | Promised\_date |
| Customer | hidden | Yes | N/A | N/A | customer | Name |
| Schedule String Sequence | hidden | Yes | N/A | N/A | caster\_string\_slab | Sched\_string\_seq |
| Schedule Heat Sequence | hidden | Yes | N/A | N/A | caster\_string\_slab | Sched\_heat\_seq |
| Slab Thickness | hidden | Yes | N/A | N/A | caster\_string\_slab | Slab\_thickness |
| Planned Slab Type | hidden | Yes | N/A | N/A | caster\_string\_slab | Planned\_slab\_type |
| Line Item Number | hidden | Yes | N/A | N/A | caster\_string\_slab | Line\_item\_no |
| Grade | hidden | Yes | N/A | N/A | caster\_string\_slab | Grade |
| Elapsed Casting time | hidden | Yes | N/A | N/A | caster\_string\_slab | Elapsed\_casting\_time |
| Elapsed Footage | hidden | Yes | N/A | N/A | caster\_string\_slab | Elapsed\_footage |
| Elapsed roll Footage | hidden | Yes | N/A | N/A | caster\_string\_slab | Elapsed\_roll\_footage |
| Elapsed String Casting Time | hidden | Yes | N/A | N/A | caster\_string\_slab | Elapsed\_string\_casting\_time |
| Slab Number | hidden | Yes | N/A | N/A | caster\_string\_slab | Slab\_no |
| Coil Length | hidden | Yes | N/A | N/A | caster\_string\_slab | Coil\_length |
| Slab Length | hidden | Yes | N/A | N/A | caster\_string\_slab | Slab\_length |
| Stock Id | hidden | Yes | N/A | N/A | caster\_string\_slab | Stock\_id |
| Slab Head Cut Time | hidden | Yes | N/A | N/A | caster\_string\_slab | Slab\_head\_cut\_time |
| Change Datetime | hidden | Yes | N/A | N/A | caster\_string\_slab | Change\_datetime |
| Change User Id | hidden | Yes | N/A | N/A | caster\_string\_slab | Change\_user\_id |
| Standar Slab Width | hidden | Yes | N/A | N/A | caster\_string\_slab | Slab\_width\_std |
| Average Slab width | hidden | Yes | N/A | N/A | caster\_string\_slab | Slab\_width\_avg |
| Transition classification | hidden | Yes | N/A | N/A | caster\_string\_slab | Transition\_classigication |
| Casting speed | hidden | Yes | N/A | N/A | caster\_string\_slab | Casting\_speed |
| Ordered Weight | hidden | Yes | N/A | N/A | Customer\_order\_line\_item | Ordered\_weight |
| Minimum coil Weight | hidden | Yes | N/A | N/A | Customer\_order\_line\_item | Min\_coil\_weight |
| Maximum coil Weigh | hidden | Yes | N/A | N/A | Customer\_order\_line\_item | Max\_coil\_weight |
| Heat Lot Order | hidden | Yes | N/A | N/A | Customer\_order\_line\_item | Heat\_lot\_order |
| No First Four Slabs Required | hidden | Yes | N/A | N/A | Customer\_order\_line\_item | No\_first\_four\_slabs\_ind |
| Production time | hidden | Yes | N/A | N/A | Coil | Prod\_time |
| Production Coil Number | hidden | Yes | N/A | N/A | Coil | Production\_coil\_no |
| Actual Ship Date | hidden | Yes | N/A | N/A | Coil | Actual\_ship\_date |
| Production Cycle | hidden | Yes | N/A | N/A | Coil | Production\_cycle |
| Coil Status | hidden | Yes | N/A | N/A | Coil | Coil\_status |
| Produced By Date | hidden | Yes | N/A | N/A | Coil | Produce\_by\_date |
| Coil Indicator | hidden | Yes | N/A | N/A | Coil | Coil\_indicator |
| Scheduled Date | hidden | Yes | N/A | N/A | Coil | Scheduled\_date |
| Coil Invoice | hidden | Yes | N/A | N/A | Coil | Coil\_invoice |
| Coil Shipped | hidden | Yes | N/A | N/A | Coil | Coil\_shipped |
| Comments | hidden | Yes | N/A | N/A | Coil | Comments |
| Number of times planned | hidden | Yes | N/A | N/A | Coil | Number\_times\_planned |
| Number of times planned comments | hidden | Yes | N/A | N/A | Coil | Number\_times\_planned\_comments |
| Mill PIW Aim | hidden | Yes | N/A | N/A | Coil | Mill\_piw\_aim |
| Processing Purchase Order Number | hidden | Yes | N/A | N/A | Coil | Processing\_po\_no |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to validate if the schedule is still sequenced (Step 7.5.1.1 BR2), the system must execute the same function used to get the available schedules in the combo box, and the selected schedule must be returned by that function.
* If the schedule is valid (that means that the validations passed) (dlavailableschedules.cpp Line 230) then:
  + Load the schedule detail:
    - SELECT hsm\_schedule\_no, hsm\_schedule\_description, approved\_dt\_stamp, approved\_user\_id, pattern\_id, est\_prod\_time, avg\_kmw, total\_coil\_tonnage, total\_coil\_length, number\_of\_strings, number\_of\_coils, schedule\_min\_coil\_weight, schedule\_max\_coil\_weight, schedule\_min\_width, schedule\_max\_width, schedule\_min\_gauge, schedule\_max\_gauge, strings, grades, avg\_casting\_time, transitions, heat\_tonnage, coil\_tonnage, prod\_seq, prod\_status, change\_datetime, change\_user\_id, slab\_width\_calculated, number\_of\_heats FROM sb\_hsm\_schedule WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]
  + Validate that the selected schedule has its production status different to ”S”. If it is:
    - Refresh the schedules list:
      * SELECT hsm\_schedule\_no, hsm\_schedule\_description FROM sb\_hsm\_schedule WHERE prod\_status = 'S' RDER BY prod\_seq
      * Return false (finish the function)
  + Update the planning tables status:
    - UPDATE sb\_hsm\_schedule SET prod\_status ="M" change\_user\_id = [CURRENT USER] change\_datetime = getdate() WHERE hsm\_schedule\_no =[CURRENT SCHEDULE]
    - UPDATE heat\_detail SET prod\_status ="M" change\_user\_id = [CURRENT USER] change\_datetime = getdate() WHERE hsm\_schedule\_no =[CURRENT SCHEDULE]
    - UPDATE caster\_string SET prod\_status ="M" change\_user\_id = [CURRENT USER] change\_datetime = getdate() WHERE hsm\_schedule\_no =[CURRENT SCHEDULE]
    - UPDATE caster\_string\_slab SET prod\_status ="M" change\_user\_id = [CURRENT USER] change\_datetime = getdate() WHERE hsm\_schedule\_no =[CURRENT SCHEDULE]
  + Retrieve all slabs in caster\_string\_slab
    - (SELECT cu.name, css.hsm\_schedule\_no,css.sched\_slab\_seq, css.sched\_string\_seq , css.sched\_heat\_seq, css.precedes\_turn\_around, css.slab\_thickness, css.slab\_width\_head, css.slab\_width\_tail, css.casting\_time, css.planned\_slab\_type, css.slab\_width\_change\_mode, css.order\_no, css.line\_item\_no, css.line\_item\_coil\_no, css.grade, css.mill\_width\_aim, css.mill\_gauge\_aim, css.mill\_weight\_aim, css.elapsed\_casting\_time, css.elapsed\_footage, css.elapsed\_roll\_footage, css.elapsed\_string\_casting\_time, css.slab\_no, css.prod\_status, css.coil\_length, css.slab\_length, css.stock\_id, css.slab\_head\_cut\_time, css.change\_datetime, css.change\_user\_id, css.slab\_width\_std, css.slab\_width\_avg, css.transition\_classification, css.casting\_speed, coli.ordered\_weight, coli.promised\_date, ROUND(coli.min\_coil\_weight, 0), ROUND(coli.max\_coil\_weight, 0), coli.heat\_lot\_order, coli.no\_first\_four\_slabs\_ind, c.prod\_time, c.production\_coil\_no, c.actual\_ship\_date, c.production\_cycle, c.coil\_status, c.produce\_by\_date, DATEDIFF(DAY, CONVERT(CHAR(11), GetDate(), 1), CONVERT(CHAR(11), c.produce\_by\_date, 1)), c.coil\_indicator, c.scheduled\_date, c.coil\_invoice, c.coil\_shipped, c.comments, c.number\_times\_planned, c.number\_times\_planned\_comments, c.mill\_piw\_aim, c.processing\_po\_no FROM caster\_string\_slab css,customer\_order\_line\_item coli,customer cu,coil c WHERE css.hsm\_schedule\_no =[SELECTED SCHEDULE] AND css.order\_no = coli.order\_no AND css.line\_item\_no = coli.line\_item\_no AND css.order\_no = c.order\_no AND css.line\_item\_no = c.line\_item\_no AND css.line\_item\_coil\_no = c.line\_item\_coil\_no AND coli.customer\_id = cu.customer\_id ) UNION ( SELECT NULL, /\*cu.name, \*/ css.hsm\_schedule\_no, css.sched\_slab\_seq, css.sched\_string\_seq , css.sched\_heat\_seq, css.precedes\_turn\_around, css.slab\_thickness, css.slab\_width\_head, css.slab\_width\_tail, css.casting\_time, css.planned\_slab\_type, css.slab\_width\_change\_mode, css.order\_no, css.line\_item\_no, css.line\_item\_coil\_no, css.grade, css.mill\_width\_aim, css.mill\_gauge\_aim, css.mill\_weight\_aim, css.elapsed\_casting\_time, css.elapsed\_footage, css.elapsed\_roll\_footage, css.elapsed\_string\_casting\_time, css.slab\_no, css.prod\_status, css.coil\_length, css.slab\_length, css.stock\_id, css.slab\_head\_cut\_time, css.change\_datetime, css.change\_user\_id, css.slab\_width\_std, css.slab\_width\_avg, css.transition\_classification, css.casting\_speed, NULL, /\*coli.ordered\_weight, \*/ NULL, /\*coli.promised\_date, \*/ NULL, /\*ROUND(coli.min\_coil\_weight, 0), \*/ NULL, /\*ROUND(coli.max\_coil\_weight, 0), \*/ NULL, /\*coli.heat\_lot\_order, \*/ NULL, /\*coli.no\_first\_four\_slab\_ind\*/ NULL, /\*c.prod\_time, \*/ NULL, /\*c.production\_coil\_no, \*/ NULL, /\*c.actual\_ship\_date, \*/ NULL, /\*c.production\_cycle, \*/ NULL, /\*c.coil\_status, \*/ NULL, /\*c.produce\_by\_date, \*/ 100, /\*DATEDIFF(DAY, CONVERT(CHAR(11), GetDate(), 1), CONVERT(CHAR(11), c.produce\_by\_date, 1)),\*/ NULL, /\*c.coil\_indicator, \*/ NULL, /\*c.scheduled\_date, \*/ NULL, /\*c.coil\_invoice, \*/ NULL, /\*c.coil\_shipped, \*/ NULL, /\*c.comments, \*/ NULL, /\*c.number\_times\_planned, \*/ NULL, /\*c.number\_times\_planned\_comments, \*/ NULL, /\*c.mill\_piw\_aim, \*/ NULL /\*c.processing\_po\_no \*/ FROM caster\_string\_slab css WHERE css.hsm\_schedule\_no = [SELECTED SCHEDULE] AND (css.order\_no = 0 OR css.order\_no = NULL OR css.line\_item\_no = 0 OR css.line\_item\_no = NULL OR css.line\_item\_coil\_no = 0 OR css.line\_item\_coil\_no = NULL) AND (css.stock\_id > 0 OR css.planned\_slab\_type > 0 ) )
  + Get the last slab of the list, and set its LastBeforeTundishChange property to true
  + ReSequence the HSM:
    - Define accumulators for CasterElapsedCastingTime and CasterElapsedStringCasting time, Elapsed Footage, Elapsed Roll Footage, and initialized all to 0
    - Obtain the ORIGINAL quantity of slabs for the schedule, and if its less than 0, set it to 0
    - If the original quantity of slabs is less than the actual quantity on the schedule, obtain the last slab from the list. Then if the index of that last slab is 0, set all accumulator values to 0, if not, assign the following values to the accumulators:
      * Caster Elapsed Casting Time = Elapsed casting time from the slab
      * Caster Elapsed String Casting Time = Elapsed String Casting Time from the slab
      * Elapsed Footage = Coil Length from the slab
      * Elapsed Roll Footage = Elapsed Coil Length
    - Define a new accumulator for the Total Casting time. Check if the slab precedes a roll change. If it does, the Total Casting time accumulator is set to the casting time of the slab, plus the turnaround time. If it does not precedes a roll change, the Total Casting time is set to the casting time of the slab.
    - For each of the slabs additional from the original list (7.5.1.1 BR2, Save all newly created slabs) (ie. If the list had 5 slabs and added 2 more, the additional slabs will be 6 and 7), check if it is in status “I” if its true, assign the following values:
      * Slab Head Cut Time = CasterElapsedCastingTime
      * CasterElapsedCastingTime = CasterElapsedCastingTime + Casting time of the Slab
      * Slab ElapsedCastingTime = CasterElapsedCastingTime
      * Slab ElapsedStringCastingTime = 0.0
      * ElapsedFootage = ElapsedFootage + Slab Coil Length
      * Slab Coil Length = Elapsedfootage
    - If the slab preceedsRollChange, ElapsedRollFootage=slab coil length and SlabTotalCastingtime = turn around time + slab casting time, if not, ElapsedRollFootage= ElapsedRollFootage + slab coil length and SlabTotalCastingTime = slab casting time.
      * Assign to the Mill prod slab sequence the current cycle value plus 1 (i+1)
    - Calculate the cut off points:
      * Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
        + If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
        + Set the slab as cutoff point and end the search for any more cut off points
        + If no cutoff points are located, set it to “NULL”
  + Delete MPS Heat List:
    - DELETE mps\_heat\_list
  + Save the MPS Heat List:
    - INSERT INTO mps\_heat\_list (mill\_prod\_heat\_seq, l3\_heat\_id, grade, heat\_weight,heat\_no,prod\_status,begin\_dt\_stamp, end\_dt\_stamp,change\_datetime,change\_user\_id) VALUES ([CURRENT HEAT LIST MILL SEQUENCE], [CURRENT HEAT L3 ID], [CURRENT HEAT GRADE], [CURRENT HEAT WEIGHT], [CURRENT HEAT NUMBER], [CURRENT HEAT PROD STATUS], "NULL", "NULL", getdate(), [CURRENT USER])
  + Update the next L3 coil id and heat id:
    - UPDATE mps\_parameters SET next\_l3\_heat\_id=[Next L3 Heat ID] change\_user\_id = [CURRENT USER] , change\_datetime = getdate()
    - UPDATE mps\_parameters SET next\_l3\_coil\_id=[Next L3 Coil ID] change\_user\_id = [CURRENT USER] , change\_datetime = getdate()
  + For each of the slabs on the list, save them into the slab list:
    - INSERT INTO mps\_slab\_list (mill\_prod\_slab\_seq, l3\_heat\_id, l3\_coil\_id, hsm\_schedule\_no, sched\_string\_seq, sched\_heat\_seq, sched\_slab\_seq, precedes\_turn\_around, slab\_thickness, casting\_time, order\_no, line\_item\_no, line\_item\_coil\_no, grade, mill\_width\_aim, mill\_gauge\_aim, mill\_weight\_aim, mill\_weight\_min, mill\_weight\_max, slab\_no, production\_coil\_no, prod\_status, stock\_id, slab\_width\_head, slab\_width\_tail, slab\_length, planned\_slab\_type, slab\_width\_change\_mode, coil\_length, change\_datetime, change\_user\_id, orig\_hsm\_schedule\_no, orig\_sched\_slab\_seq, slab\_width\_std, slab\_width\_avg, slab\_head\_cut\_time, elapsed\_casting\_time, percent\_string\_time\_used, elapsed\_footage, elapsed\_roll\_footage, transition\_classification, casting\_speed) VALUES ([CYCLE Mill PROD SLAB SEQ], [CYCLE L3 Heat ID], [CYCLE L3 COIL ID], [CYCLE HSM SCHED NO], [CYCLE SCHED STRING SEQ],[CYCLE SCHED HEAT SEQ], [CYCLE SCHED SLAB SEQ], [CYCLE PRECEDES TURN AROUND],[CYCLE SLAB THICKNESS],[CYCLE CASTING TIME],[CYCLE ORDER NO],[CYCLE LINE ITEM NO], [CYCLE LINE ITEM COIL NO],[CYCLE GRADE], [CYCLE MILL WIDTH AIM],[CYCLE MILL GUGE AIM]{,[C YCLE MILLWEIGHT AIM], [CYCLE MILL WEIGHT MIN],[CYCLE MILL WEIGHT MAX], [CYCLE SLAB NO], [CYCLE PRODUCTION COIL NO],[CYCLE PRODUCTION STATUS],[CYCLE STOCK ID],[CYCLE SLAB WIDTH HEAD],[CYCLE SLAB WIDTH TAIL],[CYCLE SLAB LENGTH],[CYCLE PLANNED SLAB TYPE],[SLAB WIDTH CHANGE MODE],[CYCLE COIL LENGTH],[CYCLE CHANGE DATETIME],[CYCLE CHANGE USER ID],[CYCLE ORIG HSM SCHEDULE NO],[CYCLE ORIG SCHEDULE SLAB SEQ],[CYCLE SLAB WIDTH STD],[CYCLE SLAB WIDTH AVG],[CYCLE SLAB HEAD CUT TIME],[CYCLE ELAPSED CASTING TIME],[CYCLE PCT STRING TIME USED],[CYCLE ELAPSED FOOTAGE],[CYCLE ELAPSED ROLL FOOTAGE],[CYCLE TRANSITION CLASSIFICATION], [CYCLE CASTING SPEED])
  + Send Message of all heats: (UC015)
  + Send message of all slabs (UC016)
  + If the list of the slabs where equal to 0, then send the message of all coils (UC029)
  + If not, add the slabs on the slab list after the slabs already on the list and send message for the coils added (UC030)

# UC036 – Mill Production Scheduling - Display Missed Production

## Description

This functionality displays the missed production on the Production Schedule Dashboard.

## Triggers

* When is missed production on the current schedule. It will be refreshed every 10 seconds.

## Preconditions

* The user is on the Production Schedule Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor open the Production Schedule Dashboard

## The system displays the Missed Production List on the “Missed Production” tab, with the following information (BR1)(BR2):

* Order
* Grade
* Coil Thickness
* Coil Width
* Slab Width
* Slab Weight
* Customer
* Date

## Use case ends.

## Alternative and Exception Flows

## There are no alternate flows.

## Business Rules

## BR1 – Missed Production

## Missed production is when a slab status is set to "P" prior to the last slab. Ex.: Slab 1 is set to "T". Slab 2 is set to “P” then slab 1 will miss the production, and become “Missed Production”. When this happen the slab that miss the production must be shown on the Caster/Hot Mill Slab list and will be shown in the “missed production scheduling” screen

## BR2 – Low Copper Slabs

## When the slab have a low copper specification, the background on the line in which the low copper slab is will be orange.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Order | Output | Yes | N/A |  | caster\_string\_slab | Order\_no |
| Grade | Output | Yes | N/A |  | Caster\_string\_slab | Grade |
| Coil Thickness | Output | Yes | N/A | 1 decimal place | caster\_string\_slab | Mill\_gauge\_aim |
| Coil Width | Output | Yes | N/A | 1 decimal place | caster\_string\_slab | Mill\_width\_aim |
| Slab Width | Output | Yes | N/A | 1 decimal place | caster\_string\_slab | Slab\_width\_tail |
| Slab Weight | Output | Yes | N/A | 1 decimal place | caster\_string\_slab | Slab\_weight |
| Customer | Output | Yes | N/A |  | customer | name |
| Date | Output | Yes | N/A | MM/DD/YYYY | Customer\_order\_line\_item | Promised\_date |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* To apply low cupper slabs:
  + Check for field low\_cu on table customer\_order\_line\_item,if the value is null, then it is NOT a low cupper slab. If it is not null, then mark it as low cupper slab, and highlight the whole line with a “cupper” color.
* To get the list of slabs that missed production (DLGMissedProduction.cpp line 86)
  + Check for Caster Cutoff Point (GetCutoffPoint())
  + If the caster cutoff point is less equal to -1, then do not refresh.
  + If the cutoff point is different than -1, then
    - Get the slab list (the actual list for production), and for each of the slabs on the list:
      * Validate if the position on the list is less than or equal to the cutoff point, and the slab status is equal to “T”, or the slab status is equal to “P”. If it is:
        + Validate if the slab is not planned type normal (property from the slab), and the slab production status is equal to “T”, or production status is “P”. If it is:

Add the slab to the missed production list

* + - * + If the validation failed:

Add the slab to the produced transition slab list

* + - * Validate if the slab is a low CU:
        + Assign low CU to paint background on cupper color.
* Calculate the CutOff Points ( mps001view.cpp line 2224):
  + Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
  + If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
  + Set the slab as cutoff point and end the search for any more cut off points
  + If no cutoff points are located, set it to “NULL”.

# UC037 – Mill Production Scheduling - Display Order Book

## Description

This functionality allows the operator to enter specific information that need to match with the order information in order to be displayed.

## Triggers

* When the actor clicks on the “Find” button.

## Preconditions

* The user is on the Production Schedule Dashboard, and selects the “Order Book” tab.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor open the Production Schedule Dashboard and selects the “Order Book” tab.

## The system displays Order Book tab, with the following information to be filled out by the actor:

* Grade (BR3)
* Minimum Width
* Maximum Width
* Minimum Gauge
* Maximum Gauge

## The user enters the desired data to stablish the filtering conditions

## The user clicks on the “Find” button (AF1)

## The system retrieves the orders with the matching criteria (BR1) and displays the following information (BR2):

* Order
* Grade
* Coil Thickness (Gauge)
* Coil Width
* Slab Width
* Slab Weight
* Customer
* Lot
* Dte (BR3)

## End of use case.

## Alternative and Exception Flows

## AF1 – Clear Button

## The clear button is pressed and all data entered is deleted.

## Business Rules

## BR1 – Filtering Criteria

## The search results must comply with the following rules:

* Grade must be equal to the grade on the order
* Coil width on the order must be more than or equal than minimum width entered, and less than or equal than the maximum width entered.
* Coil Gauge (thickness) must be more than or equal than the minimum Gauge and less than or equal than the maximum gauge entered.

## BR2 – Sorting Results

## By default, the sorting will be by order. However, it is possible to apply sorting by any field, by clicking on the label. Click once and will be sorted ascending, click twice and will be sorted descending.

## BR3 – Grade Combo Box

## The grade field must be entered by a combo box. This combo box will be filled out with the available grades on the database.

## BR4 – Stoplight Rules

## This field must be represented by a stoplight. The three lights must be visible at all times. They will be turned on according to the following rules:

* All Lights ON: when the actual date is older or equal than the planned shipping date.
* Red light ON: when the actual date is one day before the planned shipping date.
* Yellow light ON: when the actual date is two days before the planned shipping date.
* Green light ON: when the actual date is three days or more before the planned shipping date.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Grade | Input | Yes | Empty | Must be Selected from the Grade catalog. (combo box) | coil | Grade |
| Width Maximum | Input | Yes | Empty | Must be in millimetres. Only numbers will be accepted. The data entered must be more than the stored in this database field. | coil | Mill\_width\_aim |
| Width Minimum | Input | Yes | Empty | Must be in millimetres. Only numbers will be accepted. The data entered must be less than the stored in this database field | coil | Mill\_width\_aim |
| Gauge Maximum | Input | Yes | Empty | Only numbers will be accepted. The data entered must be more than the stored in this database field | coil | Mill\_gauge\_aim |
| Gauge Minimum | Input | Yes | Empty | Only numbers will be accepted. The data entered must be less than the stored in this database field | coil | Mill\_gauge\_aim |
| Order Number | Output | Yes | Empty | N/A | coil | Order number |
| Order Grade | Output | Yes | Empty | N/A | Coil | Grade |
| Coil Thickness | Output | Yes | Empty | N/A | coil | Gauge |
| Coil Width | Output | Yes | Empty | N/A | Coil | Mill\_width\_aim |
| Slab Width | Output | Yes | Empty | N/A | Sb\_std\_slab\_widths | Std\_slab\_width |
| Slab Weight | Output | Yes | Empty | N/A | coil | Mill\_weight\_aim |
| Customer | Output | Yes | Empty | N/A | customer | Name |
| Heat/Lot | Output | Yes | Empty | N/A | Customer\_order\_line\_item | Heat\_lot\_order |
| Dte | Output | Yes | Empty | Apply BR4 | coil | Produce\_by\_date |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to obtain the Grades for the combo box, this query must be executed:
  + SELECT DISTINCT grade FROM coil where coil\_status=’CO’

# UC038 – Mill Production Scheduling - Display Requires Disposition

## Description

This functionality will show the coils that requires to be removed from the current schedule because there was not enough steel to complete the coils.

## Triggers

* When the actor clicks on the “Requires Disposition” tab.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor open the Production Schedule Dashboard and selects the “Requires Disposition” tab.

## The system displays Requires Disposition tab, with the list of the coils that requires disposition (BR1) with the following columns:

* Order Number
* Grade
* Coil Thickness
* Coil Width
* Ordered Quantity
* Produced Quantity
* To be Produced Quantity
* Coil
* Time Planned

## End of use case.

## Alternative and Exception Flows

## There are no alternative or exception flows.

## Business Rules

## BR1 – Filtering Criteria

## The search results must comply with the following rules:

* Coils must have a “RD” status to be shown on the Requires Disposition list.

## BR2 – Low CU Specification

## When the slab have a low copper specification, the background on the line in which the low copper slab is will be orange.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Order Number | Output | Yes | Empty | N/A | Coil | Order\_no |
| Order Grade | Output | Yes | Empty | N/A | coil | Grade |
| Coil Thickness | Output | Yes | Empty | N/A | coil | Mill\_gauge\_aim |
| Coil Width | Output | Yes | Empty | N/A | Coil | Mill\_width\_aim |
| Quantity Ordered | Output | Yes | Empty | N/A | Customer\_order\_line\_item | Ordered\_weight |
| Quantity Produced | Output | Yes | Empty | Where production\_coil\_no is not null | Coil\_product\_data | Sum(actual\_coil\_weight) |
| Quantity to be Produced | Output | Yes | Empty | Where coil\_status is equal to IS or CO | Coil | Sum(mill\_weight\_aim) |
| Coil | Output | Yes | Empty | N/A | coil | Count(\*) |
| Time Planned | Output | Yes | Empty | N/A | coil | Number\_of\_times\_planned |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to get the Requires Disposition list (“RD”) the following must be executed (dlgrequreddisposition.cpp line 95):
  + Load the coils:
    - Get the number of coils:
      * SELECT COUNT(\*) FROM coil, customer\_order\_line\_item, coil\_product\_data,grade WHERE customer\_order\_line\_item.order\_no = coil.order\_no AND customer\_order\_line\_item.line\_item\_no = coil.line\_item\_no AND coil.production\_coil\_no \*= coil\_product\_data.production\_coil\_no AND coil.coil\_status = 'RD' AND grade.grade = coil.grade.
    - Validate if the number of coils is bigger than 0. If it is:
      * SELECT customer\_order\_line\_item.ordered\_weight, c.order\_no, c.line\_item\_no, c.line\_item\_coil\_no, c.grade,c.prod\_time, c.production\_coil\_no, c.actual\_ship\_date, c.production\_cycle, c.coil\_status, c.produce\_by\_date, DATEDIFF(DAY, CONVERT(CHAR(11), GetDate(), 1), CONVERT(CHAR(11), c.produce\_by\_date, 1)), c.coil\_indicator, c.mill\_width\_aim, c.mill\_gauge\_aim, c.mill\_weight\_aim, c.scheduled\_date, c.coil\_invoice, c.coil\_shipped, c.comments, c.number\_times\_planned, c.number\_times\_planned\_comments, c.change\_datetime, c.change\_user\_id, c.processing\_po\_no, customer\_order\_line\_item.heat\_lot\_order, customer\_order\_line\_item.min\_coil\_weight, customer\_order\_line\_item.max\_coil\_weight, isnull(customer\_order\_line\_item.low\_cu,'N'),customer\_order\_line\_item.no\_first\_four\_slabs\_ind FROM coil c, customer\_order\_line\_item, coil\_product\_data,grade g WHERE customer\_order\_line\_item.order\_no = c.order\_no AND customer\_order\_line\_item.line\_item\_no = c.line\_item\_no AND c.production\_coil\_no \*= coil\_product\_data.production\_coil\_no AND c.coil\_status = 'RD' AND g.grade = c.grade ORDER BY c.order\_no, c.line\_item\_no, c.line\_item\_coil\_no
    - Make slabs from coils:
      * For each of the coils on the list:
        + Get the slab thickness
        + Get the average steel density
        + Get the coil length \* 1000
        + Get the slab length \* 1000
        + Get the Casting time:

If the grade of the slab is equal to the current grade:

The casting speed is the current casting speed

If not, get the casting speed for the grade:

SELECT casting\_speed FROM sb\_steel\_grade\_families ssgf,grade g WHERE ssgf.steel\_grade\_family\_cd=g.steel\_grade\_family\_cd AND g.grade = [CYCLE SLAB GRADE]

If no casting speed is found or is less or equal than zero, then set the average casting speed.

Set the found casting speed as the “current speed” for the cylcle

* + - * + Calculate the casting time with the length in meters:

Slablength/1000

* + - * + Calculate the min and max weight:

If the slab is stock:

Min weight=Millweightaim\*0.9

Max weight=millweightaim \* 1.1

* + To calculate the tonnage for Produced and TO BE PRODUCED:
    - Use this query to get the Produced tonnage:
      * SELECT ISNULL(SUM(coil\_product\_data.actual\_coil\_weight),0) FROM coil\_product\_data ,coil WHERE coil.production\_coil\_no = coil\_product\_data.production\_coil\_no AND coil.production\_coil\_no IS NOT NULL AND coil.order\_no =[Current Order] AND coil.line\_item\_no=[COIL ITEM NUMBER]
    - Use this query to get the tonnage TO BE PRODUCED:
      * SELECT ISNULL(SUM(coil.mill\_weight\_aim),0) FROM coil WHERE coil.coil\_status in ('CO','IS') AND coil.order\_no =[Current Order] AND coil.line\_item\_no=[COIL ITEM NUMBER]
    - Store the resulting values as the new value for the slab property.

# UC039 – Mill Production Scheduling - Display Stock

## Description

This functionality displays the coils that are currently in stock for the specified characteristics by the user.

## Triggers

* When the actor clicks on the “Find” button.

## Preconditions

* The user is on the Production Schedule Dashboard, and selects the “Stock” tab.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor open the Production Schedule Dashboard and selects the “Stock” tab.

## The system displays Stock tab, with the following information to be filled out by the actor:

* Grade (BR3)
* Coil Thick
* Coil Width
* Coil Weight
* Slab Width

## The user enters the desired data to stablish the filtering conditions

## The user clicks on the “Find” button

## The system retrieves the coils with the matching criteria (BR1) and displays the following information (BR2):

* Steel Design Code
* Grade
* Thick Minimum
* Thick Maximum
* Width Minimum
* Width Maximum

## End of use case.

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## BR1 – Filtering Criteria

## The search results must comply with the following rules:

* Grade must be equal to the grade on the order
* Coil width entered must be between the maximum and minimum width in the order.
* Coil Gauge (thickness) entered must be between the maximum and minimum gauge in the order.

## BR2 – Sorting Results

## By default, the sorting will be by order. However, it is possible to apply sorting by any field, by clicking on the label. Click once and will be sorted ascending, click twice and will be sorted descending.

## BR3 – Grade Combo Box

## The grade field must be entered by a combo box. This combo box will be filled out with the available grades on the database.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Fields** |
| Grade | Input | Yes | Empty | Must be Selected from the Grade catalog. (combo box) | Sb\_stock\_grades | Grade |
| Coil Thickness | Input | Yes | Empty | Must be in millimetres. Only numbers will be accepted. Data entered must be between this two values | Sb\_stock | From\_gauge, to\_gauge |
| Coil Width | Input | Yes | Empty | Must be in millimetres. Only numbers will be accepted. Data entered must be between this two values | Sb\_stock | From\_width, to\_width |
| Steel Design Code | Output | Yes | Empty | N/A | Metallurgy\_steel\_design | Steel\_design\_cd |
| Grade | Output | Yes | Empty | N/A | Sb\_stock | grade |
| Thickness Maximum | Output | Yes | Empty | N/A | Sb\_stock | To\_gauge |
| Thickness Minimum | Output | Yes | Empty | N/A | Sb\_stock | From\_gauge |
| Width Maximum | Output | Yes | Empty | N/A | Sb\_stock | From\_width |
| Width Minimum | Output | Yes | Empty | N/A | Sb\_stock | To\_width |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to obtain the grade combo box data, the following query must be used:
  + SELECT DISTINCT grade FROM sb\_stock\_grades

# UC040 – Mill Production Scheduling - Display Casters

## Description

This functionality displays the slabs to be produced on the current schedule.

## Triggers

* When the actor clicks on the “Casters” button.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor open the Production Schedule Dashboard and selects the “Casters” tab.

## The system retrieves the slabs to be produced with the matching criteria (BR1) (BR2) (BR4) and displays the following information:

* L3 Coil Id
* HSM Sequence
* Order
* Slab Width Head
* Slab Width Standard
* Slab Width Tail
* Production Status
* Width Change
* Tundish Change
* Date Original

## End of use case.

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## BR1 – Filtering Criteria

## The search results must comply with the following rules:

* Slabs must be assigned to the schedule that is currently running.

## BR2 – Sorting Results

## By default, the sorting will be by HSM Sequence.

## BR3 – Stoplight Rules

## This field must be represented by a stoplight. The three lights must be visible at all times. They will be turned on according to the following rules:

* All Lights ON: when the actual date is older or equal than the planned shipping date.
* Red light ON: when the actual date is one day before the planned shipping date.
* Yellow light ON: when the actual date is two days before the planned shipping date.
* Green light ON: when the actual date is three days or more before the planned shipping date.

## BR4 – Low Copper Slabs

* When the slab have a low copper specification, the background on the line in which the low copper slab is will be orange.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| L3 Coil Id | Output | Yes | Empty | N/A | Mps\_slab\_list | L3\_coil\_id |
| HSM Sequence | Output | Yes | Empty | N/A | Mps\_slab\_list | Sched\_slab\_seq |
| Order | Output | Yes | Empty | N/A | Mps\_slab\_list | Order\_no |
| Slab Width Head | Output | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_head |
| Slab Width Std | Output | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_std |
| Slab Width Tail | Output | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_tail |
| Production Status | Output | Yes | Empty | N/A | Mps\_slab\_list | Prod\_status |
| Width Change | Output | Yes | Empty | If there is a width change on the slab, this field will be checked | Mps\_slab\_list | Slab\_width\_Change\_mode |
| Tundish Change | Output | Yes | Empty | If the slab is the last of the schedule, this field will be checked | Mps\_slab\_list | Precedes\_turn\_around |
| Date Original | Output | Yes | Empty | Apply BR3 | Customer\_order\_line\_item | Promised\_date |
| Customer Name | hidden | Yes | Empty | N/A | customer | Name |
| Mill Produced Slab Sequence | hidden | Yes | Empty | N/A | Mps\_slab\_list | Mill\_prod\_slab\_seq |
| L3 Heat Id | hidden | Yes | Empty | N/A | Mps\_slab\_list | L3\_heat\_id |
| HSM Schedule Number | hidden | Yes | Empty | N/A | Mps\_slab\_list | Hsm\_schedule\_no |
| Schedule String Sequence | hidden | Yes | Empty | N/A | Mps\_slab\_list | Sched\_string\_seq |
| Schedule Heat Sequence | hidden | Yes | Empty | N/A | Mps\_slab\_list | Ched\_heat\_seq |
| Slab Thickness | hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_thickness |
| Casting Time | hidden | Yes | Empty | N/A | Mps\_slab\_list | Casting\_time |
| Line Item Number | hidden | Yes | Empty | N/A | Mps\_slab\_list | Line\_item\_no |
| Line Item Coil Number | hidden | Yes | Empty | N/A | Mps\_slab\_list | Line\_item\_coil\_no |
| Grade | hidden | Yes | Empty | N/A | Mps\_slab\_list | Grade |
| Mill Width Aim | hidden | Yes | Empty | N/A | Mps\_slab\_list | Mill\_width\_aim |
| Mill Gauge Aim | hidden | Yes | Empty | N/A | Mps\_slab\_list | Mill\_gauge\_aim |
| Mill Weight Aim | hidden | Yes | Empty | N/A | Mps\_slab\_list | Mill\_weight\_aim |
| Mill weight Min | hidden | Yes | Empty | N/A | Mps\_slab\_list | Mill\_weigth\_min |
| Mill Weight Max | hidden | Yes | Empty | N/A | Mps\_slab\_list | Mill\_weight\_max |
| Slab Number | hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_no |
| Production Coil Number | hidden | Yes | Empty | N/A | Mps\_slab\_list | Production\_coil\_no |
| Stock Id | hidden | Yes | Empty | N/A | Mps\_slab\_list | Stock\_id |
| Slab Length | hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_length |
| Planned Slab Type | hidden | Yes | Empty | N/A | Mps\_slab\_list | Planned\_slab\_type |
| Slab Width Change Mode | hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_change\_mode |
| Coil Length | hidden | Yes | Empty | N/A | Mps\_slab\_list | Coil\_length |
| Change Datetime | hidden | Yes | Empty | N/A | Mps\_slab\_list | Change\_datetime |
| Change User Id | hidden | Yes | Empty | N/A | Mps\_slab\_list | Change\_user\_id |
| Original HSM Schedule Number | hidden | Yes | Empty | N/A | Mps\_slab\_list | Orig\_hsm\_schedule\_no |
| Original Schedule Slab Sequence | hidden | Yes | Empty | N/A | Mps\_slab\_list | Orig\_sched\_slab\_seq |
| Slab Width Standar | hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_std |
| Slab Width Average | hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_avg |
| Slab Head Cut time | hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_head\_cut\_time |
| Elapsed Casting Time | hidden | Yes | Empty | N/A | Mps\_slab\_list | Elapsed\_casting\_time |
| Percent String Time Used | hidden | Yes | Empty | N/A | Mps\_slab\_list | Percent\_string\_time\_used |
| Elapsed Footage | hidden | Yes | Empty | N/A | Mps\_slab\_list | Elapsed\_footage |
| Elapsed Roll Footage | hidden | Yes | Empty | N/A | Mps\_slab\_list | Elapsed\_roll\_footage |
| Transition Classification | hidden | Yes | Empty | N/A | Mps\_slab\_list | Transition\_classification |
| Casting Speed | hidden | Yes | Empty | N/A | Mps\_slab\_list | Casting\_speed |
| Ordered Weight | hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Ordered\_weight |
| Promised Date | hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Promised\_date |
| Min Coil Weight | hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Min\_coil\_weight |
| Max Coil Weight | hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Max\_coil\_weight |
| Heat Lot Order | hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Heat\_lot\_order |
| No first four slab Indicator | hidden | Yes | Empty | N/A | Customer\_order\_line\_item | No\_first\_four\_slabs\_ind |
| Low Cupper | hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Low\_cu |
| Production Time | hidden | Yes | Empty | N/A | Coil | Prod\_time |
| Actual Ship Date | hidden | Yes | Empty | N/A | Coil | Actual\_ship\_date |
| Production Cycle | hidden | Yes | Empty | N/A | Coil | Production\_cycle |
| Coil Status | hidden | Yes | Empty | N/A | Coil | Coil\_status |
| Produce By Date | hidden | Yes | Empty | N/A | Coil | Produced\_by\_date |
| Coil Indicator | hidden | Yes | Empty | N/A | Coil | Coil\_indicator |
| Scheduled Date | hidden | Yes | Empty | N/A | Coil | Scheduled\_date |
| Coil Invoice | hidden | Yes | Empty | N/A | Coil | Coil\_invoice |
| Coil Shipped | hidden | Yes | Empty | N/A | Coil | Coil\_shipped |
| Comments | hidden | Yes | Empty | N/A | Coil | Comments |
| Number Times Planned | hidden | Yes | Empty | N/A | Coil | Number\_times\_planned |
| Number Times Planned Comments | hidden | Yes | Empty | N/A | Coil | Number\_times\_planned\_comments |
| Mill PIW Aim | hidden | Yes | Empty | N/A | Coil | Mill\_piw\_aim |
| Processing Purchase Order Number | hidden | Yes | Empty | N/A | Coil | Processing\_po\_no |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* To apply low cupper slabs:
  + Check for field low\_cu on table customer\_order\_line\_item,if the value is null, then it is NOT a low cupper slab. If it is not null, then mark it as low cupper slab, and highlight the whole line with a “cupper” color.
* For each of the slabs returned:
  + Get the average steel density, and assign it to the slab property
  + Add the slab to the list box
  + Accumulate the L3 Coild id in a comma separated string
* With the accumulated string, execute the following query to obtain the Slab Violations:
  + SELECT l3\_coil\_id,violation\_no,severity,violation\_desc,rule\_id FROM mps\_slab\_violations WHERE l3\_coil\_id IN ( [ACCUMULATED COILS] )
  + Add each slab of the result of the query to the Violations List

# UC041 – Mill Production Scheduling - Display Hot Strip Mill Schedule

## Description

This functionality displays the coils to be produced on the current schedule.

## Triggers

* When the actor clicks on the “Hot Strip Mill Schedule” button.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor open the Production Schedule Dashboard and selects the “Hot Strip Mill Schedule” tab.

## The system retrieves the coils to be produced with the matching criteria (BR1) (BR2) (BR4) and displays the following information:

* HSM Sequence
* Order
* Coil Thick
* Coil Width
* Coil Weight
* Production Status
* Width Change
* Tundish Change
* Special Instruction
* Date Origin

## End of use case.

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## BR1 – Filtering Criteria

## The search results must comply with the following rules:

* Coils must be assigned to the schedule that is currently running.

## BR2 – Sorting Results

## By default, the sorting will be by HSM Sequence.

## BR3 – Stoplight Rules

## This field must be represented by a stoplight. The three lights must be visible at all times. They will be turned on according to the following rules:

* All Lights ON: when the actual date is older or equal than the planned shipping date.
* Red light ON: when the actual date is one day before the planned shipping date.
* Yellow light ON: when the actual date is two days before the planned shipping date.
* Green light ON: when the actual date is three days or more before the planned shipping date.

## BR4 - Low Copper Coil

* When the coil have a low copper specification, the background on the line in which the low copper coil is will be orange.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| HSM Sequence | Output | Yes | Empty | N/A | Mps\_slab\_list | Sched\_slab\_seq |
| Order | Output | Yes | Empty | N/A | Mps\_slab\_list | Order\_no |
| Coil Thick | Output | Yes | Empty | N/A | Mps\_slab\_list | Mill\_gauge\_aim |
| Coil Width | Output | Yes | Empty | N/A | Mps\_slab\_list | Mill\_width\_aim |
| Mill Weight Aim | Output | Yes | Empty | N/A | Mps\_slab\_list | Mill\_weight\_aim |
| Production Status | Output | Yes | Empty | N/A | Mps\_slab\_list | Prod\_status |
| Width Change | Output | Yes | Empty | If there is a width change on the slab, this field will be checked | Mps\_slab\_list | Slab\_width\_Change\_mode |
| Tundish Change | Output | Yes | Empty | If the slab is the last of the schedule, this field will be checked | Mps\_slab\_list | Precedes\_turn\_around |
| Number Times Planned Comments | Output | Yes | Empty | N/A | Coil | Number\_times\_planned\_comments |
| Date Original | Output | Yes | Empty | Apply BR3 | Customer\_order\_line\_item | Promised\_date |
| L3 Coil Id | Hidden | Yes | Empty | N/A | Mps\_slab\_list | L3\_coil\_id |
| Slab Width Head | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_head |
| Slab Width Std | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_std |
| Slab Width Tail | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_tail |
| Customer Name | Hidden | Yes | Empty | N/A | customer | Name |
| Mill Produced Slab Sequence | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Mill\_prod\_slab\_seq |
| L3 Heat Id | Hidden | Yes | Empty | N/A | Mps\_slab\_list | L3\_heat\_id |
| HSM Schedule Number | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Hsm\_schedule\_no |
| Schedule String Sequence | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Sched\_string\_seq |
| Schedule Heat Sequence | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Ched\_heat\_seq |
| Slab Thickness | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_thickness |
| Casting Time | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Casting\_time |
| Line Item Number | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Line\_item\_no |
| Line Item Coil Number | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Line\_item\_coil\_no |
| Grade | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Grade |
| Mill weight Min | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Mill\_weigth\_min |
| Mill Weight Max | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Mill\_weight\_max |
| Slab Number | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_no |
| Production Coil Number | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Production\_coil\_no |
| Stock Id | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Stock\_id |
| Slab Length | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_length |
| Planned Slab Type | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Planned\_slab\_type |
| Slab Width Change Mode | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_change\_mode |
| Coil Length | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Coil\_length |
| Change Datetime | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Change\_datetime |
| Change User Id | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Change\_user\_id |
| Original HSM Schedule Number | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Orig\_hsm\_schedule\_no |
| Original Schedule Slab Sequence | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Orig\_sched\_slab\_seq |
| Slab Width Standar | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_std |
| Slab Width Average | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_width\_avg |
| Slab Head Cut time | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Slab\_head\_cut\_time |
| Elapsed Casting Time | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Elapsed\_casting\_time |
| Percent String Time Used | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Percent\_string\_time\_used |
| Elapsed Footage | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Elapsed\_footage |
| Elapsed Roll Footage | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Elapsed\_roll\_footage |
| Transition Classification | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Transition\_classification |
| Casting Speed | Hidden | Yes | Empty | N/A | Mps\_slab\_list | Casting\_speed |
| Ordered Weight | Hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Ordered\_weight |
| Promised Date | Hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Promised\_date |
| Min Coil Weight | Hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Min\_coil\_weight |
| Max Coil Weight | Hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Max\_coil\_weight |
| Heat Lot Order | Hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Heat\_lot\_order |
| No first four slab Indicator | Hidden | Yes | Empty | N/A | Customer\_order\_line\_item | No\_first\_four\_slabs\_ind |
| Low Cupper | Hidden | Yes | Empty | N/A | Customer\_order\_line\_item | Low\_cu |
| Production Time | Hidden | Yes | Empty | N/A | Coil | Prod\_time |
| Actual Ship Date | Hidden | Yes | Empty | N/A | Coil | Actual\_ship\_date |
| Production Cycle | Hidden | Yes | Empty | N/A | Coil | Production\_cycle |
| Coil Status | Hidden | Yes | Empty | N/A | Coil | Coil\_status |
| Produce By Date | Hidden | Yes | Empty | N/A | Coil | Produced\_by\_date |
| Coil Indicator | Hidden | Yes | Empty | N/A | Coil | Coil\_indicator |
| Scheduled Date | Hidden | Yes | Empty | N/A | Coil | Scheduled\_date |
| Coil Invoice | Hidden | Yes | Empty | N/A | Coil | Coil\_invoice |
| Coil Shipped | Hidden | Yes | Empty | N/A | Coil | Coil\_shipped |
| Comments | Hidden | Yes | Empty | N/A | Coil | Comments |
| Number Times Planned | Hidden | Yes | Empty | N/A | Coil | Number\_times\_planned |
| Mill PIW Aim | Hidden | Yes | Empty | N/A | Coil | Mill\_piw\_aim |
| Processing Purchase Order Number | Hidden | Yes | Empty | N/A | Coil | Processing\_po\_no |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* To apply low cupper slabs:

Check for field low\_cu on table customer\_order\_line\_item,if the value is null, then it is NOT a low cupper slab. If it is not null, then mark it as low cupper slab, and highlight the whole line with a “cupper” color.

# UC042 - Mill Production Scheduling - Display Caster Graph

## Description

This functionality displays caster graph that compares the thickness and width variation of the slabs in the schedule.

## Triggers

* When the actor clicks on the “Caster Graph” tab.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor open the Production Schedule Dashboard and selects the “Caster Graph” tab.

## The system retrieves the coils to be produced with the matching criteria (BR1) and draws a chart (BR2) width the width and thickness of each slab of the current schedule.

## End of use case.

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## BR1 – Filtering Criteria

## The search results must comply with the following rules:

* Slabs must be assigned to the schedule that is currently running.

## BR2 – Chart Axis Drawing

## The chart will have two “Y” axis, width, and thickness. The “X” axis will be each of the slabs on the current schedule. Slab numbers on the X axis will not be shown.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Slab Thick | Output | Yes | Empty | N/A | Caster\_string\_slab | Slab\_gauge\_aim |
| Slab Width | Output | Yes | Empty | N/A | Caster\_string\_slab | Slab\_width\_aim |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* The chart has these characteristics:
  + Both y-axis (thickness and width) will be represented as lines.
  + Thickness will be a blue line
  + Width will be a red line.
  + When a slab is selected on the schedule, the chart will point where is located, with some form of highlighting that the library provides.

# UC043 - Mill Production Scheduling - Display Hot Strip Mill Graph

## Description

This functionality displays caster graph that compares the thickness and width variation of the coils in the schedule.

## Triggers

* When the actor clicks on the “Display Hot Strip Mill Graph” tab.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor open the Production Schedule Dashboard and selects the “Display Hot Strip Mill Graph” tab.

## The system retrieves the coils to be produced with the matching criteria (BR1) and draws a chart (BR2) width the width and thickness of each coil of the current schedule.

## End of use case.

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## BR1 – Filtering Criteria

## The search results must comply with the following rules:

* Coils must be assigned to the schedule that is currently running.

## BR2 – Chart Axis Drawing

## The chart will have to “Y” axis, width, and thickness. The “X” axis will be each of the coils on the current schedule. Coil numbers on the X axis will not be shown.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Tables** | **Database Fields** |
| Coil Thick | Output | Yes | Empty | N/A | Caster\_string\_slab | Mill\_gauge\_aim |
| Coil Width | Output | Yes | Empty | N/A | Caster\_string\_slab | Mill\_width\_aim |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

* The chart has these characteristics:
  + Both y-axis (thickness and width) will be represented as lines.
  + Thickness will be a blue line
  + Width will be a red line.

When a slab is selected on the schedule, the chart will point where is located, with some form of highlighting that the library provides.

# UC044 - Mill Production Scheduling - Display Workspace

## Description

This functionality display the slabs that are at the Workspace at the moment.

## Triggers

* When the actor clicks on the “Workspace” tab.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor open the Production Schedule Dashboard and selects the “Workspace” tab.

## The system retrieves the slabs that are at the Workspace.

## End of use case.

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## No business rules required for this use case.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Fields** |
| L3 Coil Id | Output | Yes | Empty | N/A | WORKSPACE | L3\_COIL\_ID |
| Coil Grade | Output | Yes | Empty | N/A | WORKSPACE | COIL\_GRADE |
| Coil Thick | Output | Yes | Empty | N/A | WORKSPACE | COIL\_THICK |
| Coil Width | Output | Yes | Empty | N/A | WORKSPACE | COIL\_WIDTH |
| Slab Width | Output | Yes | Empty | N/A | WORKSPACE | SLAB\_WITH |
| Slab Weight | Output | Yes | Empty | N/A | WORKSPACE | SLAB\_WEIGHT |
| Customer | Output | Yes | Empty | N/A | WORKSPACE | CUSTOMER |
| Promised Date | Output | Yes | Empty | N/A | WORKSPACE | PROM\_DATE |
| Origin | Output | Yes | Empty | N/A | WORKSPACE | ORIGIN |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

* The table for the workspace must be created.

# UC045 - Mill Production Scheduling – Save

## Description

This functionality allow to the user save the current setting of the schedule and missed production.

## Triggers

* When the actor clicks on the “Save” button.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## The actor clicks on the “Save” button (BR1)(BR2)(BR3)(BR4).

## End of use case.

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## BR1 – Save Sequencing

## The sequencing of the coils and slabs is stored as is at the moment of clicking the button.

## BR2 – Save Missed Production

## The coils that are currently on the missed production are stored with the “Missed Production” status.

## BR3 – Send Message to L2

## The PDI Message (ADC004) is sent to level 2 (Using UC0015, UC0016).

## BR4 – Check for Slabs on requires disposition

## Validate that there are no slabs on the require disposition status. If there is one or more, those must be deleted from the coil list for the caster, and update the status to “RD” on the coil.

## BR5 – Check for Slabs on Stock

## Validate that there are no slabs on the stock tab. If there is one or more, those must be deleted from the coil list for the caster.

## BR6 – Recalculate “To be Produced” and “Produced” fields

## The tonnes to be produced and tonnes already produced must be recalculated and updated.

## BR7 – Save MPS heat list and update all Slabs

## The heat list have to be rebuilt and all slabs updated (reload them from the system).

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Fields** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* These are the actions to be performed by the system in order to SAVE (SQL.CPP Line 1698) the changes done in the schedule:
  + Begin a transaction. If any of the steps fails, all changes have to be rolled back.
  + Delete all the slabs in the table mps\_slab\_list that where modified
    - Obtain all modified slabs
    - Delete the slabs from the table mps\_slab\_list that where modified. This are the slabs that where changed status in the dashboard.
  + Insert all the modified slabs ( the newly created slabs ) into the mps\_slab\_list table
  + For each of the slabs on the list, validate if it has been modified.If it is, check the origin of the modified slabs, and set their coils. The table to be updated is “COIL”, and set the coil\_status to “RA”
    - UPDATE coil SET coil\_status= 'RA', change\_datetime = getdate(), change\_user\_id = [CURRENT USER] WHERE order\_no=[CURRENT ORDER] AND line\_item\_no=[LINE ITEM OF CURRENT ITEM ON LOOP] AND line\_item\_coil\_no=[LINE COIL ITEM OF CURRENT ITEM ON LOOP].
  + Check for slabs in Requires disposition (SQL.CPP Line 1798):
    - For each or the slabs on the required disposition list:
      * Validate if the slab has been modified. If it is:
        + Delete it from the string:

DELETE caster\_string\_slab WHERE order\_no = [CURRENT SLAB ORDER] AND line\_item\_no = [CURRENT SLAB LINE ITEM] AND line\_item\_coil\_no = [CURRENT LINE ITEM COIL NO]

* + - * + Update the coil status:

UPDATE coil SET coil\_status = 'RD', change\_datetime = getdate(), change\_user\_id = [CURRENT USER] WHERE order\_no = [CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM NUMBER] AND line\_item\_coil\_no = [CURRENT LINE ITEM COIL NO]

* + Check for slabs in STOCK (SQL.CPP Line 1845):
    - Obtain all slabs in STOCK
    - Delete each of these slabs from the caster\_string\_slab table which its schedule number property is not null or empty (this means that they were part of a schedule so they have to be removed).
      * DELETE caster\_string\_slab WHERE hsm\_schedule\_no = [CURRENT SLAB SCHEDULE] AND sched\_slab\_seq = [CURRENT SLAB SCHEDULE SEQ] AND sched\_string\_seq = [CURRENT SLAB SCHEDULE STRING SEQ] AND sched\_heat\_seq = [CURRENT SLAB SCHEDULE HEAT SEQ]
  + To recalculate the tonnage for Produced and TO BE PRODUCED:
    - Use this query to get the Produced tonnage:
      * SELECT ISNULL(SUM(coil\_product\_data.actual\_coil\_weight),0) FROM coil\_product\_data ,coil WHERE coil.production\_coil\_no = coil\_product\_data.production\_coil\_no AND coil.production\_coil\_no IS NOT NULL AND coil.order\_no =[Current Order] AND coil.line\_item\_no=[COIL ITEM NUMBER]
    - Use this query to get the tonnage TO BE PRODUCED:
      * SELECT ISNULL(SUM(coil.mill\_weight\_aim),0) FROM coil WHERE coil.coil\_status in ('CO','IS') AND coil.order\_no =[Current Order] AND coil.line\_item\_no=[COIL ITEM NUMBER]
    - Store the resulting values as the new value for the slab property.
  + Build the heat list
  + Reset all slabs to not modified: mark all slabs status to not modified
  + Send Message of all heats: (UC015)
  + Send message of all slabs (UC016)
  + For each of the slabs, send a coil added message (UC030) when the coil number for the slabs, is less than the next coil number.

**In order to build the heats for the selected schedule:**

* Create a variable to accumulate all tons for the heat
* Get the first slab to calculate:
  + Obtain the current slab list
  + For each of the slabs, validate if the status is different than empty and different than”I” and the planned type is normal
  + If it is, get the heat list, and validate for each heat if the heat id from the slab, is the same than the heat from the heat list.
  + If it is, return the heat
* Calculate the CutOff Points:
  + Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
  + If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
  + Set the slab as cutoff point and end the search for any more cut off points
  + If no cutoff points are located, set it to “NULL”.
* Validate if the heat is different to “Null”, if it is, then assign to the next L3 heat id, the current L3 heat id + 1
* Obtain the first slab of the first heat that is not being produced at the time (heat not locked)
* Delete all heats that are not linked to any slabs in the schedule from the heat list from the current schedule
* Resequence all heats that are still on the list, ordered by sequence
* Recalculate Heat Boundaries:
  + Get the first and last slab for the current heat
  + Validate for each of the slabs starting from the last up to the first, if the heat from the first slab is the same as the one on the slab being validated. If it is different, it finishes the validation and stores on a variable that slab as the current slab
  + Then check that each of the slabs are not on the Missed Production list, if they are, they must be removed from the slab list
  + Accumulate the tonnage for the slabs. For each of the slabs these validations will be performed:
    - Validate if the slab is the first after a turnaround or is the las before tundish change
    - If the slab is the first of the list, or is the last before a tundish change, the yield loss head has to be added to the actual kg accumulated
    - If not, add the mill\_weight\_aim of the slab
    - If the slab is the last of the schedule or is a turnaround, the yield loss tail has to be accumulated. Additional to the weight.
  + Assign the current heat to the slab
  + Then validate if:
    - There is too much kg (accumulated kg is more than the aim size of the heat)
    - There is a grade change (grade of the current slab compared against the grade of the one evaluated and the quantity of the slabs are less than the calculated initially)
  + If any of these is false, accumulate the kg, and validate the following slab.
* Slabs that are on the missed production should be removed.
* After slabs that are not needed are removed, the remaining list must be evaluated as follows:
  + For each of the slabs:
    - Create a variable to accumulate the kg and set it to 0
    - Create a variable to set if the slab is the last from the list or before tundish change.
    - Validate if it’s the last slab from the list, or if it’s the last before tundish change, if it is, set the variable to true, if not, set to false.
    - Validate if the slab is the first of the list or is the last before tundish change. If it is, add to the accumulated kg the yield loss head multiplied by the kg per metric ton property of the slab.
    - Add the mill\_weight\_aim of the slab to the accumulated variable
    - If is turn around or end of schedule, then add the yield loss tail multiplied by the kg per metric ton property, to the accumulator variable.
    - Add the property Heat to the slab with the current heat.
    - Validate if the total accumulated kg plus the delta accumulation is more than the aimheatsize and set the result in a variable for TooMuchKg
    - Validate if the grade for the last slab is the same as the current slab, and store the result.
* If there is too much kg, a grade change or a turnaround or end of schedule, a new heat is created:
  + Heat sequence is incremented by one from the last heat
  + Heat Id from L3 is incremented by one from the last heat
  + Grade is obtained from the slab grade of the current slab
  + Weight is obtained using these rules:
    - If too much kg and is not a turnaround or end of schedule, then heat weight is the aim heat size multiplied by the kg per metric ton factor. If the heat also is a grade change, the weight will be the aim heat size multiplied by the kg per metric ton plus the accumulated kg (totalaccumulatedkg = accumulatedfromheat - (aim heat size \* kg per metric ton).
    - If too much kg and is a turnaround or end of schedule: the heat weight will be the totalaccumulatedkg plus the accumulated from the heat.
    - If its anything else then the heat weight will be the totalaccumulatedkg plus the accumulated from the heat.

Boundaries of the heat must have graphical representation (like the brackets)

# UC046 - Mill Production Scheduling - Print

## Description

This functionality generate a printer friendly version of the schedule.

## Triggers

* When the actor clicks on the “Print” button.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the tab for “Casters” (AF1)

## The actor clicks on the “Print” button (BR1)

## The system generates a printer friendly version of the schedule on a modal window so it can be printed through the web browser.

## End of use case.

## Alternative and Exception Flows

## AF1 – Select Hot Strip Mill Schedule

## The user selects the “Hot Strip Mill Schedule”

## The flow returns to step 18.5.1.2.

## Business Rules

## BR1 – Information to be Printed

## The information to be printed will be specified by the tab that is active at the time, with the same layout as in the screen.

## BR2 – Stoplight Rules

## This field must be represented by a stoplight. The three lights must be visible at all times. They will be turned on according to the following rules:

* All Lights ON: when the actual date is older or equal than the planned shipping date.
* Red light ON: when the actual date is one day before the planned shipping date.
* Yellow light ON: when the actual date is two days before the planned shipping date.
* Green light ON: when the actual date is three days or more before the planned shipping date.

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| L3 Coil Id | Output | Yes | Empty | Printed in Caster | CASTER\_STRING\_SLAB | LINE\_ITEM\_COIL\_NO |
| HSM Sequence | Output | Yes | Empty | Printed in Caster and Hot strip | CASTER\_STRING\_SLAB | SCHED\_SLAB\_SEQ |
| Order | Output | Yes | Empty | Printed in Caster and Hot strip | CASTER\_STRING\_SLAB | ORDER\_NO |
| Slab Width Head | Output | Yes | Empty | Printed in Caster | CASTER\_STRING\_SLAB | SLAB\_WIDTH\_HEAD |
| Slab Width Standard | Output | Yes | Empty | Printed in Caster | CASTER\_STRING\_SLAB | SLAB\_WIDTH\_STD |
| Slab Width Tail | Output | Yes | Empty | Printed in Caster | CASTER\_STRING\_SLAB | SLAB\_WIDTH\_TAIL |
| Production Status | Output | Yes | Empty | Printed in Caster and Hot strip | CASTER\_STRING\_SLAB | PROD\_STATUS |
| Width Change | Output | Yes | Empty | Printed in Caster and Hot strip | CASTER\_STRING\_SLAB | SLAB\_WIDTH\_CHANGE\_MODE |
| Tardish Change | Output | Yes | Empty | Printed in Caster and Hot strip | CASTER\_STRING\_SLAB | PRECEDES\_TURN\_AROUND |
| Date Origin | Output | Yes | Empty | Printed in Caster and Hot strip. Apply BR2 | CUSTOMER\_ORDER\_LINE\_ITEM | PROMISED\_DATE |
| Coil Thick | Output | Yes | Empty | Printed in Hot strip | MPS\_SLAB\_LIST | MILL\_GAUGE\_AIM |
| Coil Width | Output | Yes | Empty | Printed in Hot strip | MPS\_SLAB\_LIST | MILL\_WIDTH\_AIM |
| Coil Weight | Output | Yes | Empty | Printed in Hot strip | MPS\_SLAB\_LIST | MILL\_WEIGHT\_AIM |
| Special Instructions | Output | Yes | Empty | Printed in Hot Strip. If the coil has an special instruction, it will show a triangle icon with an exclamation mark. | COIL | PROCESSING\_PO\_NO |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

* In order to perform the printing, the system will display a emergent window with the content of the selected, then the user will have to use the printing option from the browser.

# UC047 - Mill Production Scheduling - Refresh

## Description

This functionality allow to refresh the data displayed on the Production Schedule Dashboard.

## Triggers

* When the actor clicks on the “Refresh” button.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Refresh” button (BR1)

## The system executes UC035 – Obtain Schedule

## The system executes UC036 – Display Missed Production

## The system executes UC037 – Display Order Book

## The system executes UC038 – Display Requires Disposition

## The system executes UC039 – Display Stock.

## The system executes UC040 – Display Casters

## The system executes UC041 – Display Hot Strip Mill Schedule

## The system executes UC042 - Display Caster Graph

## The system executes UC043 - Display Hot strip Mill Graph

## The system executes UC044 - Display workspace

## The system executes UC057 – Auto Release Schedule .

## End of use case.

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules required in this use case.

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

The system must execute the same functions as the ones used on the use cases mentioned on the main flow.

There must be a parameter in which specifies how often the system will be automatically refreshed and execute this use case.

# UC048 - Mill Production Scheduling - Heat List

## Description

This functionality displays the heats associated with the current schedule.

## Triggers

* When the actor clicks on the “Heat List” button.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Heat List” button

## The system calculates the cut off points

## The system recalculates the heats

## The system re sequences the heats

## The system calculates the heat boundaries.

## The system opens a modal window with the heat list for the current schedule, and the following columns (BR1) (AF1):

* + Production Sequence
  + Level 3 Heat Id
  + Production Status
  + Heat Number L2
  + Grade
  + Weight

## End of use case.

## Alternative and Exception Flows

## Click on the “Alternate Grades” button:

## The user clicks on the “Alternate Grades” button

## The system executes UC0054

## Use case ends.

## Business Rules

## BR1 – Filtered Information

## The heat list will be filtered by the current schedule. The list contains the heats that are assigned to the schedule.

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Tables** | **Database Fields** |
| Production Spec | Output | Yes | Empty | Increment by one from the mill\_prod\_heat\_seq of the current slab | N/A | N/A |
| Level 3 Heat Id | Output | Yes | Empty | Increment by one from the l3\_heat\_id of the current slab | N/A | N/A |
| Production Status | Output | Yes | Empty | Set to empty | N/A | N/A |
| Heat Number L2 | Output | Yes | Empty | Set to NULL | N/A | N/A |
| Grade | Output | Yes | Empty | Grade from the M\_grade field | N/A | N/A |
| Weight | Output | Yes | Empty | The accumulated weight | N/A | N/A |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

In order to obtain the heats for the selected schedule:

* Create a variable to accumulate all tons for the heat
* Get the first slab to calculate:
  + Obtain the current slab list
  + For each of the slabs, validate if the status is different than empty and different than ”I” and Is Planned Type == “Normal”
  + If it is, get the heat list, and validate for each heat if the heat id from the slab, is the same than the heat from the heat list.
  + If it is, return the heat
* Calculate the CutOff Points:
  + Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
  + If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
  + Set the slab as cutoff point and end the search for any more cut off points
  + If no cutoff points are located, set it to “NULL”.
* If the heat is different than “NULL” , then assign to the next L3 heat id, the current heat + 1.
* Obtain the first slab of the first heat that is not being produced at the time (heat not locked)
* Delete all heats that are not linked to any slabs in the schedule from the heat list from the current schedule
* Resequence all heats that are still on the list, ordered by sequence
* Recalculate Heat Boundaries:
  + Get the first and last slab for the current heat
  + Validate for each of the slabs starting from the last up to the first, if the heat from the first slab is the same as the one on the slab being validated. If it is different, it finishes the validation and stores on a variable that slab as the current slab
  + Then check that each of the slabs are not on the Missed Production list, if they are, they must be removed from the slab list
  + Accumulate the tonnage for the slabs. For each of the slabs these validations will be performed:
    - Validate if the slab is the first after a turnaround or is the las before tundish change
    - If the slab is the first of the list, or is the last before a tundish change, the yield loss head has to be added to the actual kg accumulated
    - If not, add the mill\_weight\_aim of the slab
    - If the slab is the last of the schedule or is a turnaround, the yield loss tail has to be accumulated. Additional to the weight.
  + Assign the current heat to the slab
  + Then validate if:
    - There is too much kg (accumulated kg is more than the aim size of the heat)
    - There is a grade change (grade of the current slab compared against the grade of the one evaluated and the quantity of the slabs are less than the calculated initially)
  + If any of these is false, accumulate the kg, and validate the following slab.
* Slabs that are on the missed production should be removed.
* After slabs that are not needed are removed, the remaining list must be evaluated as follows:
  + For each of the slabs:
    - Create a variable to accumulate the kg and set it to 0
    - Create a variable to set if the slab is the last from the list or before tundish change.
    - Validate if it’s the last slab from the list, or if it’s the last before tundish change, if it is, set the variable to true, if not, set to false.
    - Validate if the slab is the first of the list or is the last before tundish change. If it is, add to the accumulated kg the yield loss head multiplied by the kg per metric ton property of the slab.
    - Add the mill\_weight\_aim of the slab to the accumulated variable
    - If is turn around or end of schedule, then add the yield loss tail multiplied by the kg per metric ton property, to the accumulator variable.
    - Add the property Heat to the slab with the current heat.
    - Validate if the total accumulated kg plus the delta accumulation is more than the aimheatsize and set the result in a variable for TooMuchKg
    - Validate if the grade for the last slab is the same as the current slab, and store the result.
* If there is too much kg, a grade change or a turnaround or end of schedule, a new heat is created:
  + Heat sequence is incremented by one from the last heat
  + Heat Id from L3 is incremented by one from the last heat
  + Grade is obtained from the slab grade of the current slab
  + Weight is obtained using these rules:
    - If too much kg and is not a turnaround or end of schedule, then heat weight is the aim heat size multiplied by the kg per metric ton factor. If the heat also is a grade change, the weight will be the aim heat size multiplied by the kg per metric ton plus the accumulated kg (totalaccumulatedkg = accumulatedfromheat - (aim heat size \* kg per metric ton).
    - If too much kg and is a turnaround or end of schedule: the heat weight will be the totalaccumulatedkg plus the accumulated from the heat.
    - If its anything else then the heat weight will be the totalaccumulatedkg plus the accumulated from the heat.
* Boundaries of the heat must have graphical representation (like the brackets)

# UC049 - Mill Production Scheduling - PDI Update / Retransmit PDI

## Description

This functionality allows to resend the PDI information to L2. It could be used when production of an existing schedule is aborted; if the production of the current schedule can be continued, the operator does not have anything to do with the system unless the Level 2 Computer systems require that the PDI be resent.

## Triggers

* When the actor clicks on the “Retransmit PDI” button.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Retransmit PDI” button

## The system executes UC015 – Send all Heats

## The system executes UC016 – Send all Slabs

## The system executes UC029 – Send all Coils

## End of use case.

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules required for this use case.

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

The system must use the same processing logic as implemented in the use cases for UC015, UC016, UC029.

UC050 - Mill Production Scheduling - Width Verification – Not Used

## Description

This functionality allows check the slab width.

## Triggers

* When the actor clicks on the “Check Width” button.

## Preconditions

* The user is on the Production Schedule Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Transition Slab” button

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules for this use case.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

None

# UC051 - Mill Production Scheduling -Transition Slab – Not Used

## Description

This functionality allows to clear the transition slab.

## Triggers

* When the actor clicks on the “Transition Slab” button.

## Preconditions

* The user is on the Production Schedule Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Transition Slab” button

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules for this use case.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

None

UC052 - Mill Production Scheduling -Delays – Not Used

## Description

This functionality allows to enter the delays on the system. This is the delays related to the MPS, so its not used.

## Triggers

* When the actor clicks on the “Transition Slab” button.

## Preconditions

* The user is on the Production Schedule Dashboard

## Actors

* User

## Basic Flow

## The actor clicks on the “Transition Slab” button

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules for this use case.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

None

# UC053 - Mill Production Scheduling – Display Alternate Grade

## Description

This functionality allows to display the alternate grades for the slab or slabs selected from the current schedule, and assign an alternate grade if needed.

## Triggers

* When the actor clicks on the “Alternate Grade” button.

## Preconditions

* The user is on the Production Schedule Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor select the coil or coils of which wants to know their alternate grades.

## The actor clicks on the “Alternate Grade” button

## The system shows the alternate grade list that are available for the selected coils (BR1) on a modal window.

## End of use case

## Alternative and Exception Flows

## AF1 – Click on the “Cancel” button.

## The user clicks on the Cancel Button

## The system closes the Alternate Grade Window

## End use case.

## Business Rules

## BR1 – List of Alternate Grades

## The list of alternate grades that are shown must be available for all selected slabs. If one of the alternate grades is not available for one of the slab, it will not be shown.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Selected | Input | Yes | Not Selected | The current grade will be selected. In order to save, one grade have to be selected. | Metallurgy\_steel\_design | grade |
| Alternate Grade | Output | Yes | N/A | N/A | Metallurgy\_steel\_design | Alt\_grade\_1, Alt\_grade\_2, Alt\_grade\_3, Alt\_grade\_4, Alt\_grade\_5 |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* To get the list of alternate grades for the selected slab:
  + If it is not a stock slab use this query: SELECT msd.steel\_design\_cd, msd.grade, msd.alt\_grade\_1, msd.alt\_grade\_2, msd.alt\_grade\_3, msd.alt\_grade\_4, msd.alt\_grade\_5 FROM customer\_order\_line\_item coli, metallurgy\_steel\_design msd WHERE coli.order\_no = [SELECTED ORDER] AND coli.line\_item\_no = [SELECTED COIL] AND coli.steel\_design\_cd = msd.steel\_design\_cd
  + If it is a stock slab, use this query: SELECT msd.steel\_design\_cd, msd.grade, msd.alt\_grade\_1, msd.alt\_grade\_2, msd.alt\_grade\_3, msd.alt\_grade\_4, msd.alt\_grade\_5 FROM sb\_stock s, metallurgy\_steel\_design msd WHERE s.stock\_id = [SELECTED ORDER] AND coli.line\_item\_no = [SELECTED COIL] AND s.steel\_design\_cd = msd.steel\_design\_cdç

# UC054 - Mill Production Scheduling - Select Alternate Grade

## Description

This functionality allows to assign an alternate grade if needed from the heat list.

## Triggers

* When the actor clicks on the “Alternate Grade” button.

## Preconditions

* The user is on the Production Schedule Dashboard, and the heat list modal window is open.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor select the heat of which wants to know their alternate grades.

## The actor clicks on the “Alternate Grade” button

## The system shows the alternate grade list that are available for the selected heat on a modal window.

## The user selects one alternate grade from the list

## The user click on the “Save” button (AF1).

## The system updates the grades for the selected heat and closes the modal window (BR1).

## End of use case

## Alternative and Exception Flows

## AF1 – Click on the “Cancel” button.

## The user clicks on the Cancel Button

## The system closes the Alternate Grade Window

## End use case.

## Business Rules

## BR1 – Actions to perform to update steel grades

## The Heat from the schedule must be rebuilt

## Missed Production Caster and Mill grids must be updated.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
| Selected | Input | Yes | Not Selected | The current grade will be selected. In order to save, one grade have to be selected. |
| Alternate Grade | Output | Yes | N/A | N/A |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* When the user selects an alternate grade and clicks on save:
  + First, get the slabs selected
  + If the grade selected is the same as the grade on the slab, then do nothing.
  + If the grade selected is different as the grade on the slab:
    - Set the grade of the slab to the new grade selected
    - Refresh the missed production, caster list and mill list.
* Build the heats from the schedule (explained after ending this process)
* Delete the Heat List from MPS
  + DELETE mps\_heat\_list
* Save the new MPS Heat List
  + Insert each of the heats on the list built before:
    - INSERT INTO mps\_heat\_list (mill\_prod\_heat\_seq, l3\_heat\_id, grade, heat\_weight,heat\_no,prod\_status,begin\_dt\_stamp, end\_dt\_stamp,change\_datetime,change\_user\_id) VALUES ([CURRENT HEAT LIST MILL SEQUENCE], [CURRENT HEAT L3 ID], [CURRENT HEAT GRADE], [CURRENT HEAT WEIGHT], [CURRENT HEAT NUMBER], [CURRENT HEAT PROD STATUS], "NULL", "NULL", getdate(), [CURRENT USER])

**Build Heats List From Schedule**

In order to build the heats for the selected schedule (Function: BuildHeatsFromSchedule):

* Create a variable to accumulate all tons for the heat
* Get the first slab to calculate:
  + Obtain the current slab list
  + For each of the slabs, validate if the status is different than empty and different than”I” and the planned type is normal
  + If it is, get the heat list, and validate for each heat if the heat id from the slab, is the same than the heat from the heat list.
  + If it is, return the heat
* Calculate the CutOff Points:
  + Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
  + If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
  + Set the slab as cutoff point and end the search for any more cut off points
  + If no cutoff points are located, set it to “NULL”.
* Validate if the heat is different to “Null”, if it is, then assign to the next L3 heat id, the current L3 heat id + 1
* Obtain the first slab of the first heat that is not being produced at the time (heat not locked)
* Delete all heats that are not linked to any slabs in the schedule from the heat list from the current schedule
* Resequence all heats that are still on the list, ordered by sequence
* Recalculate Heat Boundaries:
  + Get the first and last slab for the current heat
  + Validate for each of the slabs starting from the last up to the first, if the heat from the first slab is the same as the one on the slab being validated. If it is different, it finishes the validation and stores on a variable that slab as the current slab
  + Then check that each of the slabs are not on the Missed Production list, if they are, they must be removed from the slab list
  + Accumulate the tonnage for the slabs. For each of the slabs these validations will be performed:
    - Validate if the slab is the first after a turnaround or is the las before tundish change
    - If the slab is the first of the list, or is the last before a tundish change, the yield loss head has to be added to the actual kg accumulated
    - If not, add the mill\_weight\_aim of the slab
    - If the slab is the last of the schedule or is a turnaround, the yield loss tail has to be accumulated. Additional to the weight.
  + Assign the current heat to the slab
  + Then validate if:
    - There is too much kg (accumulated kg is more than the aim size of the heat)
    - There is a grade change (grade of the current slab compared against the grade of the one evaluated and the quantity of the slabs are less than the calculated initially)
  + If any of these is false, accumulate the kg, and validate the following slab.
* Slabs that are on the missed production should be removed.
* After slabs that are not needed are removed, the remaining list must be evaluated as follows:
  + For each of the slabs:
    - Create a variable to accumulate the kg and set it to 0
    - Create a variable to set if the slab is the last from the list or before tundish change.
    - Validate if it’s the last slab from the list, or if it’s the last before tundish change, if it is, set the variable to true, if not, set to false.
    - Validate if the slab is the first of the list or is the last before tundish change. If it is, add to the accumulated kg the yield loss head multiplied by the kg per metric ton property of the slab.
    - Add the mill\_weight\_aim of the slab to the accumulated variable
    - If is turn around or end of schedule, then add the yield loss tail multiplied by the kg per metric ton property, to the accumulator variable.
    - Add the property Heat to the slab with the current heat.
    - Validate if the total accumulated kg plus the delta accumulation is more than the aimheatsize and set the result in a variable for TooMuchKg
    - Validate if the grade for the last slab is the same as the current slab, and store the result.
* If there is too much kg, a grade change or a turnaround or end of schedule, a new heat is created:
  + Heat sequence is incremented by one from the last heat
  + Heat Id from L3 is incremented by one from the last heat
  + Grade is obtained from the slab grade of the current slab
  + Weight is obtained using these rules:
    - If too much kg and is not a turnaround or end of schedule, then heat weight is the aim heat size multiplied by the kg per metric ton factor. If the heat also is a grade change, the weight will be the aim heat size multiplied by the kg per metric ton plus the accumulated kg (totalaccumulatedkg = accumulatedfromheat - (aim heat size \* kg per metric ton).
    - If too much kg and is a turnaround or end of schedule: the heat weight will be the totalaccumulatedkg plus the accumulated from the heat.
    - If its anything else then the heat weight will be the totalaccumulatedkg plus the accumulated from the heat.

Boundaries of the heat must have graphical representation (like the brackets)

UC055 - Mill Production Scheduling - Check Schedule – Not Used

## Description

This functionality allows to check the schedule. It is not used.

## Triggers

* When the actor clicks on the “Check Schedule” button.

## Preconditions

* The user is on the Production Schedule Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Schedule” button

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules for this use case.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

None

# UC056 - Mill Production Scheduling - Release Schedule

## Description

This functionality allows to release the selected schedule. When production of an existing schedule is aborted; if the production of the current schedule can’t be continued, the operator will have to release the current schedule and obtain a new one. To release the current schedule, the operator needs to use the Release Schedule function. Select the schedule to release.

## Triggers

* When the actor clicks on the “Release Schedule” button.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Release Schedule” button

## The system opens a modal window with the schedules available to release

## The user selects one of the schedules on the list an clicks on the “Release” button (AF1)

## The system displays the detail of the selected schedule.

## The user confirms the release (AF1).

## The system releases the schedule selected. (BR1),

## The system saves the changes using UC045

## The system sends the messages for all heats and slabs (UC015, UC016)

## The system closes the modal window.

## End of use case

## Alternative and Exception Flows

## AF1 - The user clicks on the “Cancel” Button

## The user clicks on the “Cancel” Button

## The system closes the modal window

## End of use case.

## Business Rules

## BR1 – Coils to be released

## Only the coils with an status “P” (Produced) can be released from the schedule

## The remaining coils that where not produced changes its status to “RD” requires disposition.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to release the current schedule, the following actions must be performed (DLGRELEASE.CPP Line 105):
  + Load the selected schedule information:
    - SELECT hsm\_schedule\_no, hsm\_schedule\_description,approved\_dt\_stamp, approved\_user\_id,pattern\_id, est\_prod\_time, avg\_kmw, total\_coil\_tonnage, total\_coil\_length,number\_of\_strings,number\_of\_coils, schedule\_min\_coil\_weight, schedule\_max\_coil\_weight, schedule\_min\_width, schedule\_max\_width, schedule\_min\_gauge, schedule\_max\_gauge, strings,grades, avg\_casting\_time, transitions,heat\_tonnage, coil\_tonnage,prod\_seq, prod\_status, change\_datetime, change\_user\_id, slab\_width\_calculated ,number\_of\_heats FROM sb\_hsm\_schedule WHERE hsm\_schedule\_no =[SELECTED SCHEDULE]
  + Validate if there is a slab in production, noSlabAtAll and notmodified schedule:
    - Obtain the total of slab on the current schedule
    - Validate for each of the slabs from the cutoff point up to the last slab on the schedule, if the schedule number is the same as the current schedule, if it is, set the variable noSlabAtAll to false.
    - If the slab status is equal to “T” or to “C”, set the noslabinproduction variable to false.
    - If there were the same number of coils in the current schedule as in the slabs on the schedule, set the notmodified to false.
    - If any of these three variables is true, a message informing that the schedule is invalid will be displayed.
  + Display the schedule detail, and the user still can cancel at this point. If its confirmed, then:
    - Start a SQL transaction
    - If the schedule does not have any other associated slabs (noSlabAtAll is set to true) then:
      * UPDATE sb\_hsm\_schedule SET prod\_status = 'A', change\_user\_id = [CURRENT USER], change\_datetime = getdate() WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]
      * UPDATE heat\_detail SET prod\_status = 'A', change\_user\_id = [CURRENT USER], change\_datetime = getdate() WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]
      * UPDATE caster\_string SET prod\_status = 'A', change\_user\_id = [CURRENT USER], change\_datetime = getdate() WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]
      * UPDATE caster\_string\_slab SET prod\_status = 'A', change\_user\_id = [CURRENT USER], change\_datetime = getdate() WHERE hsm\_schedule\_no = [SELECTED SCHEDULE]
    - If there is no slab in production (noslabinproduction==true) and the schedule has not been modified (bNotModified==true) then execute the following queries:
      * delete from caster\_string\_slab where hsm\_schedule\_no= [CURRENT SCHEDULE]
      * delete from caster\_string where hsm\_schedule\_no = [CURRENT SCHEDULE]
      * delete from heat\_detail where hsm\_schedule\_no = [CURRENT SCHEDULE]
      * delete from sb\_hsm\_schedule where hsm\_schedule\_no = [CURRENT SCHEDULE]
    - Else, if any of the noslabinproduction, bnotmodified or noslabatall is false, then:
      * Move unproduced coils to disposition:
        + Get the coils from the current schedule:

SELECT css.order\_no, css.line\_item\_no, css.line\_item\_coil\_no, c.coil\_status FROM caster\_string\_slab css, coil c WHERE css.hsm\_schedule\_no = [CURRENT SCHEDULE] AND css.prod\_status <> 'P' AND c.coil\_status = 'IS' AND css.order\_no = c.order\_no AND css.line\_item\_no = c.line\_item\_no AND css.line\_item\_coil\_no = c.line\_item\_coil\_no

* + - * + For each of the slabs, set its status to “RD”:

UPDATE coil SET coil\_status = 'RD', change\_datetime = getdate(), change\_user\_id = [USER] WHERE order\_no = [SLAB ORDER]AND line\_item\_no =[SLAB ITEM NO] AND line\_item\_coil\_no = [SLAB COIL NO]

Trigger the Stored Procedure to update the line item status:

UPDATE customer\_order\_line\_item SET order\_no=order\_no, change\_datetime = getdate(), change\_user\_id = [USER] WHERE order\_no=[SLAB ORDER] AND line\_item\_no=[SLAB ITEM]

* + - * + Delete the schedule from planning tables:

delete from caster\_string\_slab where hsm\_schedule\_no= [CURRENT SCHEDULE]

delete from caster\_string where hsm\_schedule\_no = [CURRENT SCHEDULE]

delete from heat\_detail where hsm\_schedule\_no = [CURRENT SCHEDULE]

delete from sb\_hsm\_schedule where hsm\_schedule\_no = [CURRENT SCHEDULE]

* + - * Find the minimum indice for the schedule slabs:
        + For each of the slab on the list, starting from the top, validate if the schedule of the slab is the same as the current schedule. If it is, set the minimum for the HSM sequence as the index number.
      * Delete the unproduced slabs from the list:
        + DELETE mps\_slab\_list WHERE hsm\_schedule\_no=[CURRENT SCHEDULE]
    - Resequence the HSM (MPS001View.cpp Line 1359) (explained after this process)
    - Build the heat list and save it:
      * Delete the content of the mps\_heat\_list table
        + DELETE mps\_heat\_list
      * Insert each of the heats on the list built before:
        + INSERT INTO mps\_heat\_list (mill\_prod\_heat\_seq, l3\_heat\_id, grade, heat\_weight,heat\_no,prod\_status,begin\_dt\_stamp, end\_dt\_stamp,change\_datetime,change\_user\_id) VALUES ([CURRENT HEAT LIST MILL SEQUENCE], [CURRENT HEAT L3 ID], [CURRENT HEAT GRADE], [CURRENT HEAT WEIGHT], [CURRENT HEAT NUMBER], [CURRENT HEAT PROD STATUS], "NULL", "NULL", getdate(), [CURRENT USER])
    - Save all changes (Same functionality as described on UC045 will be executed at this point)
  + Commit SQL transaction
  + Send the Messages from UC015 and UC016
  + Refresh the requires disposition grid.
  + Reset all al slabs pDirty parameter to false.
  + Refresh HotStrip Mill and Caster Graphs.

**Re Sequence HSM**

* To resequence the schedule it must perform the following:
  + Define accumulators for CasterElapsedCastingTime and CasterElapsedStringCasting time, Elapsed Footage, Elapsed Roll Footage, and initialized all to 0
  + Obtain the ORIGINAL quantity of slabs for the schedule, and if its less than 0, set it to 0
  + If the original quantity of slabs is less than the actual quantity on the schedule, obtain the last slab from the list. Then if the index of that last slab is 0, set all accumulator values to 0, if not, assign the following values to the accumulators:
    - Caster Elapsed Casting Time = Elapsed casting time from the slab
    - Caster Elapsed String Casting Time = Elapsed String Casting Time from the slab
    - Elapsed Footage = Coil Length from the slab
    - Elapsed Roll Footage = Elapsed Coil Length
  + Define a new accumulator for the Total Casting time. Check if the slab precedes a roll change. If it does, the Total Casting time accumulator is set to the casting time of the slab, plus the turnaround time. If it does not precedes a roll change, the Total Casting time is set to the casting time of the slab.
  + For each of the slabs additional from the original list (7.5.1.1 BR2, Save all newly created slabs) (ie. If the list had 5 slabs and added 2 more, the additional slabs will be 6 and 7), check if it is in status “I” if its true, assign the following values:
    - Slab Head Cut Time = CasterElapsedCastingTime
    - CasterElapsedCastingTime = CasterElapsedCastingTime + Casting time of the Slab
    - Slab ElapsedCastingTime = CasterElapsedCastingTime
    - Slab ElapsedStringCastingTime = 0.0
    - ElapsedFootage = ElapsedFootage + Slab Coil Length
    - Slab Coil Length = Elapsedfootage
    - If the slab preceedsRollChange, ElapsedRollFootage=slab coil length and SlabTotalCastingtime = turn around time + slab casting time, if not, ElapsedRollFootage= ElapsedRollFootage + slab coil length and SlabTotalCastingTime = slab casting time.
    - Assign to the Mill prod slab sequence the current cycle value plus 1 (i+1)
  + Calculate the cut off points:
    - Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
      * If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
      * Set the slab as cutoff point and end the search for any more cut off points

If no cutoff points are located, set it to “NULL”.

# UC057 - Mill Production Scheduling - AutoRelease Schedule

## Description

This functionality allows to release the coils that have been from the actual schedule.

## Triggers

* When the actor clicks on the “AutoRelease Schedule” button
* When the next produced slab is the first for the heat.

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Release Schedule” button

## The system will remove the coils that have already been produced (BR1)

## End of use case

## Alternative and Exception Flows

## There are no alternative or exception flows.

## Business Rules

## BR1 – Coils to be released

## Only the coils with an status “P” (Produced) can be released from the schedule

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* To process an auto release the following Stored procedure needs to be executed:
  + DECLARE @need\_refresh INT , @error\_flag INT , @error\_msg VARCHAR(255) EXECUTE spc\_mps\_auto\_release [GUID] , @need\_refresh OUTPUT , @error\_flag OUTPUT , @error\_msg OUTPUT
  + A status bar must be shown to let the user know that the system is doing something
  + After a autorelease, all objects on the screen must be refreshed.

UC058 - Mill Production Scheduling - Release Slab – Not Used

## Description

This functionality allows to Release a single slab. It is not used.

## Triggers

* When the actor clicks on the “Release Slab” button.

## Preconditions

* The user is on the Production Schedule Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Release Slab” button

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules for this use case.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

None

# UC059 - Mill Production Scheduling - Suspend Interfaces

## Description

This functionality allows to suspend the messages coming from Level 2 so the user can update his schedule without being affected by the updates.

## Triggers

* When the actor clicks on the “Suspend Interfaces” button

## Preconditions

* The user is on the Production Schedule Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Suspend Interfaces” button

## The system will change the status of the interface processing. (BR1)

## End of use case

## Alternative and Exception Flows

## There are no alternative or exception flows.

## Business Rules

## BR1 – Change Status of Interface Processing

## When the interfaces are active, the status must be changed to inactive

## When the interfaces are inactive, the status must be changed to active

## There must be a visual indicator of the status.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* If the global parameter is set to suspend interfaces, then all messages must be received, but not processed.

# UC060 - Mill Production Scheduling - Add/ Remove pieces to Schedule – Not Used

## Description

This functionality allows to check the schedule. It is not used.

## Triggers

* When the actor clicks on the “Check Schedule” button.

## Preconditions

* The user is on the Production Schedule Dashboard

## Actors

* User

## Basic Flow

## The actor clicks on the “Schedule” button

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules for this use case.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

None

# UC061 - Mill Production Scheduling - Display Piece Detail

## Description

This functionality allows the user to get the Piece Detail of the selected item.

## Triggers

* When the actor double clicks the item to be displayed.

## Preconditions

* The user is on the Production Schedule Dashboard

## Actors

* User

## Basic Flow

## The actor double clicks on the item that want to know its detail

## The system opens a modal window with the following information:

## Order Information

## Customer

## Order Line Item

## Number of Coil

## Quality

## Mill External

## Caster Internal

## Caster External

## Dates

## Produced By

## General

## Grade

## Heat Lot

## Special Instruction

## Number of times planned

## Slab Length

## Casting Time

## Aspect Ratio

## Measurements

## Slab thick

## Slab width

## Coil Weight Aim

## Coil weight min

## Coil weight max

## Coil weight aim mill

## Coil thickness aim

## Coil thickness min

## Coil thickness max

## Coil thickness aim mill

## Coil width aim

## Coil width min

## Coil width max

## Coil width aim mill

## Planning Data

## Level 3 coil id

## HSM Schedule

## HSM Schedule description

## Casting time

## String Sequence within schedule

## Heat Sequence within schedule

## Slab sequence within schedule

## Total slabs in schedule

## Slab Width Head

## Slab Width Tail

## Order Detail

## Order Number

## Line item

## Line item coil

## Grade

## Promised date

## Steel Design Information

## Steel Design Code

## Steel Design description

## Production Data

## Status

## Order Boundary Chemistries

## Element

## Aim

## Min

## Max

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules for this use case.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Order Information - Customer | Output | Yes | Empty | Not Required | customer | Name |
| Order Information - Order Line Item | Output | Yes | Empty | Not Required | Mps\_slab\_list | Line\_item\_no |
| Order Information - Number of Coil | Output | Yes | Empty | Not Required | Mps\_slab\_list | Line\_item\_coil\_no |
| Quality - Mill External | Output | Yes | Empty | Not Required |  |  |
| Quality - Caster Internal | Output | Yes | Empty | Not Required |  |  |
| Quality - Caster External | Output | Yes | Empty | Not Required |  |  |
| Dates - Produced By | Output | Yes | Empty | Not Required | coil | Produced\_by |
| General - Grade | Output | Yes | Empty | Not Required | Mps\_slab\_list | Grade |
| General - Heat Lot | Output | Yes | Empty | Not Required | Mps\_slab\_list | L3\_heat\_id |
| General - Special Instruction | Output | Yes | Empty | Not Required | coil | Number\_times\_planned\_comments |
| General - Number of times planned | Output | Yes | Empty | Not Required | coil | Number\_times\_planned |
| General - Slab Length | Output | Yes | Empty | Not Required | Mps\_slab\_list | Slab\_length |
| General - Casting Time | Output | Yes | Empty | Not Required | Mps\_slab\_list | Casting\_time |
| General - Aspect Ratio | Output | Yes | Empty | Not Required | Mps\_slab\_list | Mill\_width\_aim/mill\_gauge\_aim |
| Measurements - Slab thick | Output | Yes | Empty | Not Required | Mps\_slab\_list | Slab\_thicnkess |
| Measurements - Slab width | Output | Yes | Empty | Not Required | Caster\_string\_slab | Slab\_width\_std |
| Measurements - Coil Weight Aim | Output | Yes | Empty | Not Required | Mps\_slab\_list | Mill\_weight\_aim |
| Measurements - Coil weight min | Output | Yes | Empty | Not Required | Customer\_order\_line\_item | Min\_coil\_weight |
| Measurements - Coil weight max | Output | Yes | Empty | Not Required | Customer\_order\_line\_item | Max\_coil\_weight |
| Measurements - Coil weight aim mill | Output | Yes | Empty | Not Required | Customer\_order\_line\_item | Ordered\_weight |
| Measurements - Coil thickness aim | Output | Yes | Empty | Not Required | Mps\_slab\_list | Mill\_gauge\_aim |
| Measurements - Coil thickness min | Output | Yes | Empty | Not Required | Rolling\_mill\_practice | Mill\_gauge\_min |
| Measurements - Coil thickness max | Output | Yes | Empty | Not Required | Rolling\_mill\_practice | Mill\_gauge\_max |
| Measurements - Coil thickness aim mill | Output | Yes | Empty | Not Required | coil | Mill\_gauge\_aim |
| Measurements - Coil width aim | Output | Yes | Empty | Not Required | Mps\_slab\_list | Mill\_width\_aim |
| Measurements - Coil width min | Output | Yes | Empty | Not Required | Rolling\_mill\_practice | Mill\_width\_min |
| Measurements - Coil width max | Output | Yes | Empty | Not Required | Rolling\_mill\_practice | Mill\_width\_max |
| Measurements - Coil width aim mill | Output | Yes | Empty | Not Required | coil | Mill\_width\_aim |
| Planning Data - Level 3 coil id | Output | Yes | Empty | Not Required | Mps\_slab\_list | L3\_coil\_id |
| Planning Data - HSM Schedule | Output | Yes | Empty | Not Required | Sb\_hsm\_schedule | Hsm\_schedule\_no |
| Planning Data - HSM Schedule description | Output | Yes | Empty | Not Required | Sb\_hsm\_schedule | Hsm\_schedule\_description |
| Planning Data - String Sequence within schedule | Output | Yes | Empty | Not Required | Caster\_string\_slab | Sched\_string\_seq |
| Planning Data - Heat Sequence within schedule | Output | Yes | Empty | Not Required | Caster\_string\_slab | Sched\_heat\_seq |
| Planning Data - Slab sequence within schedule | Output | Yes | Empty | Not Required | Caster\_string\_slab | Sched\_slab\_seq |
| Planning Data - Total slabs in schedule | Output | Yes | Empty | Obtain the slabs for the schedule in which the slab is being programmed | Caster\_string\_slab | Count(\*) from schedule\_no |
| Planning Data - Slab Width Head | Output | Yes | Empty | Not Required | Caster\_string\_slab | Slab\_width\_head |
| Planning Data - Slab Width Tail | Output | Yes | Empty | Not Required | Caster\_string\_slab | Slab\_width\_tail |
| Order Detail - Order Number | Output | Yes | Empty | Not Required | Caster\_string\_slab | Order\_no |
| Order Detail - Line item | Output | Yes | Empty | Not Required | Caster\_string\_slab | Line\_item\_no |
| Order Detail - Line item coil | Output | Yes | Empty | Not Required | Caster\_string\_slab | Line\_item\_coil\_no |
| Order Detail - Grade | Output | Yes | Empty | Not Required | Caster\_string\_slab | Grade |
| Order Detail - Promised date | Output | Yes | Empty | Not Required | Customer\_order\_line\_item | Promised\_date |
| Steel Design Information - Steel Design Code | Output | Yes | Empty | Not Required | Customer\_order\_line\_item | Steel\_design\_cd |
| Steel Design Information - Steel Design description | Output | Yes | Empty | Not Required | Metallurgy\_steel\_design | Steel\_design\_description |
| Production Data - Status | Output | Yes | Empty | Not Required | Mps\_slab\_list | Prod\_status |
| Order Boundary Chemistries - Element | Output | Yes | Empty | Displayed on a table. This will display the name of the element on the chemistries | Chemistries | Name of the Element by column |
| Order Boundary Chemistries - Aim | Output | Yes | Empty | Displayed on a table. This will display the aim quantity of the element. | Chemistries | |  | | --- | | element\_al\_sol \_aim, | | element\_al\_total\_aim, | | element\_b\_aim, | | element\_c\_aim, | | element\_ca\_aim, | | element\_cr\_aim, | | element\_cu\_aim, | | element\_mn\_aim, | | element\_mo\_aim, | | element\_n \_aim, | | element\_nb\_aim, | | element\_ni\_aim, | | element\_p\_aim, | | element\_s\_aim, | | element\_si\_aim, | | element\_sn\_aim, | | element\_ti\_aim, | | element\_v\_aim, | | element\_pb\_aim, | | element\_zn\_aim, | | element\_zr\_aim, | | element\_as\_aim, | | element\_mg\_aim, | | element\_co\_aim, | | element\_sb\_aim, | |
| Order Boundary Chemistries - Min | Output | Yes | Empty | Displayed on a table. This will display the min quantity of the element. | Chemistries | |  | | --- | | element\_al\_sol \_min, | | element\_al\_total\_min, | | element\_b\_min, | | element\_c\_min, | | element\_ca\_min, | | element\_cr\_min, | | element\_cu\_min, | | element\_mn\_min, | | element\_mo\_min, | | element\_n \_min, | | element\_nb\_min, | | element\_ni\_min, | | element\_p\_min, | | element\_s\_min, | | element\_si\_min, | | element\_sn\_min, | | element\_ti\_min, | | element\_v\_min, | | element\_pb\_min, | | element\_zn\_min, | | element\_zr\_min, | | element\_as\_min, | | element\_mg\_min, | | element\_co\_min, | | element\_sb\_min, | |
| Order Boundary Chemistries - Max | Output | Yes | Empty | Displayed on a table. This will display the max quantity of the element. | Chemistries | |  | | --- | | element\_al\_sol \_max, | | element\_al\_total\_max, | | element\_b\_max, | | element\_c\_max, | | element\_ca\_max, | | element\_cr\_max, | | element\_cu\_max, | | element\_mn\_max, | | element\_mo\_max, | | element\_n \_max, | | element\_nb\_max, | | element\_ni\_max, | | element\_p\_max, | | element\_s\_max, | | element\_si\_max, | | element\_sn\_max, | | element\_ti\_max, | | element\_v\_max, | | element\_pb\_max, | | element\_zn\_max, | | element\_zr\_max, | | element\_as\_max, | | element\_mg\_max, | | element\_co\_max, | | element\_sb\_max, | |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to execute step 33.5.1.2 display the slab detail, we need to obtain all slab information (SQL.CPP, Line 290):
  + Validate: if the slab is stock, and it is on a schedule (hsm\_schedule\_no!=”NULL” and hsm\_schedule\_no!=””)
    - If its true:
      * SELECT (SELECT COUNT(\*)

FROM caster\_string\_slab WHERE hsm\_schedule\_no=[CURRENT SCHEDULE]) as S1,shs.hsm\_schedule\_description,msd.steel\_design\_cd,msd.steel\_design\_description ,rmp.carbon\_equivalent,rmp.mill\_gauge\_min,rmp.mill\_gauge\_max,rmp.mill\_gauge\_cto\_max,rmp.mill\_gauge\_cto\_min,rmp.mill\_width\_min,rmp.mill\_width\_max,rmp.mill\_width\_cto\_max,rmp.mill\_width\_cto\_min,rmp.crown\_aim,rmp.crown\_min,rmp.crown\_max,rmp.crown\_cto\_max,rmp.crown\_cto\_min,rmp.wedge\_aim,rmp.wedge\_min,rmp.wedge\_max,rmp.wedge\_cto\_max,rmp.wedge\_cto\_min,rmp.flatness\_aim,rmp.flatness\_min,rmp.flatness\_max,rmp.flatness\_cto\_max,rmp.flatness\_cto\_min,rmp.finish\_temp\_aim,rmp.finish\_temp\_min,rmp.finish\_temp\_max,rmp.finish\_temp\_cto\_max,rmp.finish\_temp\_cto\_min,rmp.coiling\_temp\_aim,rmp.coiling\_temp\_min,rmp.coiling\_temp\_max,rmp.coiling\_temp\_cto\_min,rmp.coiling\_temp\_cto\_max

FROM sb\_hsm\_schedule shs, customer\_order\_line\_item coli, customer\_order co, customer cu, metallurgy\_steel\_design msd, rolling\_mill\_practice rmp

WHERE coli.order\_no = [CURRENT ORDER]

AND coli.line\_item\_no = [CURRENT ITEM]

AND shs.hsm\_schedule\_no = [CURRENT SCHEDULE]

AND co.order\_no =[CURRENT ORDER]

AND cu.customer\_id = co.customer\_id

AND msd.steel\_design\_cd = coli.steel\_design\_cd AND msd.rolling\_mill\_If practice\_cd = rmp.rolling\_mill\_practice\_cd

* + - Where:
      * [CURRENT ORDER] is the order in which the selected slab is assigned.
      * [CURRENT ITEM] is the item number from the selected slab.
      * [CURRENT SCHEDULE] is the schedule in which the selected slab is assigned
  + If previous validation fails, then validate if the slab is stock, and it is not on a schedule (hsm\_schedule\_no==”NULL” or hsm\_schedule\_no==””)
    - If its true:
      * SELECT msd.steel\_design\_cd,msd.steel\_design\_description ,rmp.carbon\_equivalent,rmp.mill\_gauge\_min,rmp.mill\_gauge\_max,rmp.mill\_gauge\_cto\_max,rmp.mill\_gauge\_cto\_min,rmp.mill\_width\_min,rmp.mill\_width\_max,rmp.mill\_width\_cto\_max,rmp.mill\_width\_cto\_min,rmp.crown\_aim,rmp.crown\_min,rmp.crown\_max,rmp.crown\_cto\_max,rmp.crown\_cto\_min,rmp.wedge\_aim,rmp.wedge\_min,rmp.wedge\_max,rmp.wedge\_cto\_max,rmp.wedge\_cto\_min,rmp.flatness\_aim,rmp.flatness\_min,rmp.flatness\_max,rmp.flatness\_cto\_max,rmp.flatness\_cto\_min,rmp.finish\_temp\_aim,rmp.finish\_temp\_min,rmp.finish\_temp\_max,rmp.finish\_temp\_cto\_max,rmp.finish\_temp\_cto\_min,rmp.coiling\_temp\_aim,rmp.coiling\_temp\_min,rmp.coiling\_temp\_max,rmp.coiling\_temp\_cto\_min,rmp.coiling\_temp\_cto\_max

FROM customer\_order\_line\_item coli, customer\_order co, customer cu, metallurgy\_steel\_design msd,rolling\_mill\_practice rmp

WHERE coli.order\_no =[CURRENT ORDER]

AND coli.line\_item\_no = [CURRENT ITEM]

AND co.order\_no =[CURRENT ORDER]

AND cu.customer\_id = co.customer\_id

AND msd.steel\_design\_cd = coli.steel\_design\_cd AND msd.rolling\_mill\_practice\_cd = rmp.rolling\_mill\_practice\_cd

* + - Where:
      * [CURRENT ORDER] is the order in which the selected slab is assigned.
      * [CURRENT ITEM] is the item number from the selected slab
  + If both validations fails:
    - SELECT msd.steel\_design\_cd,msd.steel\_design\_description ,rmp.carbon\_equivalent,rmp.mill\_gauge\_min,rmp.mill\_gauge\_max,rmp.mill\_gauge\_cto\_max,rmp.mill\_gauge\_cto\_min,rmp.mill\_width\_min,rmp.mill\_width\_max,rmp.mill\_width\_cto\_max,rmp.mill\_width\_cto\_min,rmp.crown\_aim,rmp.crown\_min,rmp.crown\_max,rmp.crown\_cto\_max,rmp.crown\_cto\_min,rmp.wedge\_aim,rmp.wedge\_min,rmp.wedge\_max,rmp.wedge\_cto\_max,rmp.wedge\_cto\_min,rmp.flatness\_aim,rmp.flatness\_min,rmp.flatness\_max,rmp.flatness\_cto\_max,rmp.flatness\_cto\_min,rmp.finish\_temp\_aim,rmp.finish\_temp\_min,rmp.finish\_temp\_max,rmp.finish\_temp\_cto\_max,rmp.finish\_temp\_cto\_min,rmp.coiling\_temp\_aim,rmp.coiling\_temp\_min,rmp.coiling\_temp\_max,rmp.coiling\_temp\_cto\_min,rmp.coiling\_temp\_cto\_max FROM metallurgy\_steel\_design msd,sb\_stock,rolling\_mill\_practice rmp WHERE sb\_stock.stock\_id = [STOCK ID] AND msd.steel\_design\_cd = sb\_stock.steel\_design\_cd AND msd.rolling\_mill\_practice\_cd = rmp.rolling\_mill\_practice\_cd
    - Where:
      * [STOCK ID] is the stock id from the selected slab.
  + To obtain the chemistry boundaries:
    - SELECT element\_al\_sol\_aim,element\_al\_sol\_min,element\_al\_sol\_max,element\_al\_total\_aim,element\_al\_total\_min,element\_al\_total\_max,element\_b\_aim,element\_b\_min,element\_b\_max,element\_c\_aim,element\_c\_min,element\_c\_max,element\_ca\_aim,element\_ca\_min,element\_ca\_max,element\_cr\_aim,element\_cr\_min,element\_cr\_max,element\_cu\_aim,element\_cu\_min,element\_cu\_max,element\_mn\_aim,element\_mn\_min,element\_mn\_max,element\_mo\_aim,element\_mo\_min,element\_mo\_max,element\_n\_aim,element\_n\_min,element\_n\_max,element\_nb\_aim,element\_nb\_min,element\_nb\_max,element\_ni\_aim,element\_ni\_min,element\_ni\_max,element\_p\_aim,element\_p\_min,element\_p\_max,element\_s\_aim,element\_s\_min,element\_s\_max,element\_si\_aim,element\_si\_min,element\_si\_max,element\_sn\_aim,element\_sn\_min,element\_sn\_max,element\_ti\_aim,element\_ti\_min,element\_ti\_max,element\_v\_aim,element\_v\_min,element\_v\_max,element\_pb\_aim,element\_pb\_min,element\_pb\_max,element\_zn\_aim,element\_zn\_min,element\_zn\_max,element\_zr\_aim,element\_zr\_min,element\_zr\_max,element\_as\_aim,element\_as\_min,element\_as\_max,element\_mg\_aim,element\_mg\_min,element\_mg\_max,element\_co\_aim,element\_co\_min,element\_co\_max,element\_sb\_aim,element\_sb\_min,element\_sb\_max FROM customer\_order\_line\_item,chemistries WHERE customer\_order\_line\_item.boundary\_chemistry\_id=chemistries.chemistry\_id and order\_no=[CURRENT ORDER] and line\_item\_no=[CURRENT ITEM]
    - Where:
      * [CURRENT ORDER] is the order assigned to the current order
      * [CURRENT ITEM] is the item number assigned to the selected slab.
  + Additional properties not detailed on the queries, must be obtained from the object properties.

# UC062 – Slab Re-Application – Display Slabs in the Tunnel Furnace

## Description

This functionality displays the Slabs that are in the Tunnel Furnace.

## Triggers

* When the actor enters into the Slab Re-Application Dashboard.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor enters into the Slab Re-Application Dashboard.

## The system displays the list of the Slabs on the tunnel furnace with the following information:

## Slab #

## Production Coil #

## Grade

## Customer

## Weight

## Head Width

## Tail Width

## Slab Length

## L3 Coil

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## No business rules for this use case.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Fields** |
| Slab Number | Output | Yes | 0 | When the returned value is empty, set the default value | Caster\_thin\_slab\_report | Slab\_no |
| Production Coil Number | Output | Yes | N/A | N/A | Coil\_process\_data | Production\_coil\_no |
| Grade | Output | Yes | None | When the returned value is empty, set the default value | Caster\_thin\_slab\_report | Grade |
| Customer | Output | Yes | None | When the returned value is empty, set the default value | customer | Name |
| Weight | Output | Yes | 0 | When the returned value is empty, set the default value | Caster\_thin\_slab\_report | Piece\_weight |
| Head Width | Output | Yes | 0 | When the returned value is empty, set the default value | Mps\_slab\_list | Slab\_with\_head |
| Tail Width | Output | Yes | 0 | When the returned value is empty, set the default value | Mps\_slab\_list | Slab\_width\_tail |
| Slab Length | Output | Yes | 0 | When the returned value is empty, set the default value | Mps\_slab\_list | Slab\_length |
| L3 Coil | Output | Yes | N/A | N/A | Mps\_slab\_list | L3\_coil\_id |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* Select the slab list on the furnace:
  + SELECT msl.mill\_prod\_slab\_seq, msl.l3\_coil\_id, msl.grade, FROM mps\_slab\_list msl WHERE msl.prod\_status = 'T' AND msl.planned\_slab\_type = 0
* Then for each of the slabs retrieved from the previous query, execute the following query and complement with the other tables and fields on the use case:
  + SELECT slab\_no FROM caster\_thin\_slab\_report WHERE l3\_coil\_id = + sL3CoilId + GROUP BY l3\_coil\_id HAVING produced\_dt\_stamp = max(produced\_dt\_stamp)

# UC063 – Slab Re-Application – Display Matching Committed Coils for a Slab

## Description

This functionality displays the list of the committed orders that match the selected slab.

## Triggers

* When the actor enters into the Slab Re-Application Dashboard, select a coil and then click search for matches, and select one slab of the “Slabs in the Furnace” list.

## Preconditions

* The user is on the Slab Re-Application Dashboard, and a slab is selected from the slab in tunnel furnace

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on one of the slabs on the list.

## The system displays the list of the Orders that matches the specified criteria (BR1) with the following information:

## Order #

## Line #

## # Coils

## Customer

## Weight

## Width

## Gauge

## Length

## FB

## Promised

## Caster Int

## Caster Ext

## Mill.

## The system displays the list of the Coils that are on Requires Disposition that matches the specified criteria (BR1) with the following information:

## Order #

## Line #

## # Coils

## Customer

## Weight

## Width

## Gauge

## Length

## FB

## Promised

## Caster Int

## Caster Ext

## Mill.

## The system retrieves the list of coils that complies with the entered data, showing the following columns:

## Steel Design

## Grade

## Gauge Min

## Gauge Max

## Width Min

## Width Max

## Prev Jump

## Next Jump.

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## BR1 – Search Criteria Conditions

## The order list must be filtered by this rules:

## Mill External Rating

## Coil Width

## Filler Bars

## Has weight constraints

## Grade Match

## Coils must be in “CO” (Committed) or “RD” (Requires Disposition) Status.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Order Number | Output | Yes | N/A | N/A | coil | Order\_no |
| Line Number | Output | Yes | N/A | N/A | Coil | Line\_item\_no |
| Number of Coils | Output | Yes | N/A | N/A | coil | Count(\*) |
| Customers | Output | Yes | N/A | N/A | Customer | name |
| Weight | Output | Yes | N/A | N/A | coil | Mill\_weight\_aim |
| Width | Output | Yes | N/A | N/A | coil | Mill\_width\_aim |
| Gauge | Output | Yes | N/A | N/A | coil | Mill\_gauge\_aim |
| Length | Output | Yes | N/A | N/A | coil | Mill\_length\_aim |
| FB | Output | Yes | N/A | N/A | customer\_order\_line\_item | Steel\_design\_cd |
| Promised Date | Output | Yes | N/A | N/A | customer\_order\_line\_item | Promised\_date |
| Caster Int | Output | Yes | N/A | N/A | metallurgy\_steel\_design | Piece\_internal\_rating |
| Caster Ext | Output | Yes | N/A | N/A | metallurgy\_steel\_design | Piece\_external\_rating |
| Mill | Output | Yes | N/A | N/A | grade | Steel\_grade\_family\_cd |
| Coil in RD - Order Number | Output | Yes | N/A | N/A | coil | Order\_no |
| Coil in RD - Line Number | Output | Yes | N/A | N/A | Coil | Line\_item\_no |
| Coil in RD - Number of Coils | Output | Yes | N/A | N/A | coil | Count(\*) |
| Coil in RD - Customers | Output | Yes | N/A | N/A | Customer | name |
| Coil in RD - Weight | Output | Yes | N/A | N/A | coil | Mill\_weight\_aim |
| Coil in RD - Width | Output | Yes | N/A | N/A | coil | Mill\_width\_aim |
| Coil in RD - Gauge | Output | Yes | N/A | N/A | coil | Mill\_gauge\_aim |
| Coil in RD - Length | Output | Yes | N/A | N/A | coil | Mill\_length\_aim |
| Coil in RD - FB | Output | Yes | N/A | N/A | customer\_order\_line\_item | Steel\_design\_cd |
| Coil in RD - Promised Date | Output | Yes | N/A | N/A | customer\_order\_line\_item | Promised\_date |
| Coil in RD - Caster Int | Output | Yes | N/A | N/A | metallurgy\_steel\_design | Piece\_internal\_rating |
| Coil in RD - Caster Ext | Output | Yes | N/A | N/A | metallurgy\_steel\_design | Piece\_external\_rating |
| Coil in RD - Mill | Output | Yes | N/A | N/A | grade | Steel\_grade\_family\_cd |
| Coil in Stock - Steel Design | Output | Yes | Empty | N/A | Sb\_stock | Steel\_design\_cd |
| Coil in Stock -Grade | Output | Yes | Empty | N/A | Sb\_stock | Grade |
| Coil in Stock -Gauge Min | Output | Yes | Empty | N/A | Sb\_stock | From\_grade |
| Coil in Stock -Gauge Max | Output | Yes | Empty | N/A | Sb\_stock | To\_grade |
| Coil in Stock -Width Min | Output | Yes | Empty | N/A | Sb\_stock | From\_width |
| Coil in Stock -Width Max | Output | Yes | Empty | N/A | Sb\_stock | To\_width |
| Coil in Stock -Prev Jump | Output | Yes | Empty | N/A | Empty | Empty |
| Coil in Stock -Next Jump | Output | Yes | Empty | N/A | Empty | Empty |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

To get the matching coils for the selected slab (SR001VW.CPP Line 555):

* Stop refreshing data (Timer)
* Set the searching arguments for the query (these are variables that have to be defined and to be sent as parameters):
  + Execute the query to obtain the application parameters:
    - SELECT mill\_external\_rating, width\_above,width\_below, filler\_bars, manual\_grade, grade\_flag, weight\_min\_string, weight\_max\_string, ROUND(weight\_min\_percent, 3),ROUND(weight\_max\_percent, 3) FROM nb\_reapp\_parameters WHERE id\_no = 1
  + Assign the result to the app parameters:
    - pFillerBars = filler\_bars
    - pMillExternal=mill\_external\_rating
    - pSteelDesign=manual\_grade
    - pWeightConstraints= “FALSE”
    - sWidthAbove=width\_above
    - sWidthBelow= width\_below
    - pComma=”TRUE”
* Update the screen information
* Get the selected slab, and get the corresponding coil
* Prepare the matching coils (PrepareMatch SR001VW.CPP Line 446):
  + Get the sequence of the selected slab
    - SlabSeq= sequence of the selected slab
  + Calculate the previous slab:
    - PrevSlabSeq = SlabSeq - 1
  + Calculate the next slab:
    - NextSlabSeq = SlabSeq + 1
  + Validate NextSlabSeq, if its equal to 10000, then its equal to 1
  + Validate PrevSlabSeq, if its equal to 0, then its equal to 9999
  + Get gauge aim for the next and previous slabs:
    - Previous Slab Gauge Aim: SELECT ROUND(ISNULL(mps\_slab\_list.mill\_gauge\_aim,0),2) FROM mps\_slab\_list WHERE mps\_slab\_list.mill\_prod\_slab\_seq = [PrevSlabSeq]
    - Next Slab Gauge Aim: SELECT ROUND(ISNULL(mps\_slab\_list.mill\_gauge\_aim,0),2) FROM mps\_slab\_list WHERE mps\_slab\_list.mill\_prod\_slab\_seq = [NextSlabSeq]
  + Get the gauge jump gauges:
    - Previous gauge jump gauge: SELECT ROUND(from\_gauge, 2) FROM sb\_gauge\_jump\_gauges WHERE from\_gauge <= [PrevMillGaugeAim] AND [PrevMillGaugeAim] < to\_gauge OR ([PrevMillGaugeAim]<= to\_gauge AND last\_row = 'Y'))
  + Get Grades:
    - Previous Slab Grade = SELECT grade FROM mps\_slab\_list WHERE mill\_prod\_slab\_seq = [PrevSlabSeq]
    - Next Slab Grade = SELECT grade FROM mps\_slab\_list WHERE mill\_prod\_slab\_seq = [NextSlabSeq]
  + Get steel grade family:
    - Previous Slab Steel Family: SELECT steel\_grade\_family\_ctg FROM grade G, sb\_steel\_grade\_families SGF WHERE G.steel\_grade\_family\_cd = SGF.steel\_grade\_family\_cd AND G.grade = [Previous Grade]
    - Next Slab Steel Family: SELECT steel\_grade\_family\_ctg FROM grade G, sb\_steel\_grade\_families SGF WHERE G.steel\_grade\_family\_cd = SGF.steel\_grade\_family\_cd AND G.grade = [Next Grade]
* Select the Matching Coils:
  + Obtain the casterThinSlabReport grade for the selected slab
    - Table Caster\_thin\_slab\_report field grade
  + Obtain the Piece Internal Rating
    - Table Caster\_thin\_slab\_report field piece\_internal\_rating
  + Obtain the Piece External Rating
    - Table Caster\_thin\_slab\_report field piece\_external\_rating
* The query to obtain its formed dynamically. Depending on the parameters and values of the data obtained in the first step.
  + Validate if Mill external Rating parameter is different to empty. If it is:
    - **MillExternalQuery**=”AND coli.mill\_external>= [MillExternalParameter]”
  + If not:
    - **MillExternalQuery**=””
  + Validate the Grade Flag if its equal to 1. If it is:
    - **GradeQuery**=” (grade = [GradeParameter] or alt\_grade\_1= [GradeParameter] or alt\_grade\_2= [GradeParameter] or alt\_grade\_3= [GradeParameter] or alt\_grade\_4= [GradeParameter] or alt\_grade\_5= [GradeParameter])”
  + If it Is not:
    - **GradeQuery**=” (grade = [casterThinSlabReport Grade] or alt\_grade\_1= [casterThinSlabReportGrade] or alt\_grade\_2= [casterThinSlabReportGrade] or alt\_grade\_3= [casterThinSlabReportGrade] or alt\_grade\_4= [casterThinSlabReport Grade] or alt\_grade\_5= [casterThinSlabReport Grade])”
  + Weight Query:
    - **WeightQuery**= “ AND [ThinSlabReportWeight]>=[coli.WeightMinStringParam ]\* [WeightMinPercent] AND [ThinSlabReportWeight]<=[coli.WeightMaxStringParam ]\* [WeightMaxPercent]
  + Width Query (this just calculates the values that will be integrated to the query):
    - **WidthMinQuery**= slabwidth – slabWidthMin
    - **WidthMaxQuery** = slabwidth + sabWidthMax
  + Exclude Filler Bar Query, validate if the parameter GetFillerBars is equal to 0. If it is:
    - **ExcludeFillerBarQuery**= “ AND coli.order\_no = lip.order\_no AND coli.line\_item\_no = lip.line\_item\_no AND (lip.price\_override\_reason\_cd <> 9" OR OR lip.price\_override\_reason\_cd IS NULL) and sFrom = “ , Cust\_order\_line\_item\_price lip”
* Use the following dynamic query:
  + SELECT coli.order\_no, coli.line\_item\_no, ROUND(mill\_gauge\_aim, 2), boundary\_chemistry\_id, ROUND(mill\_width\_aim, 1), testing\_reqmnt\_cd FROM customer\_order\_line\_item coli, metallurgy\_steel\_design msd " + sFrom + " WHERE coli.steel\_design\_cd = msd.steel\_design\_cd AND" + **GradeQuery** + " AND EXISTS (SELECT \* FROM coil WHERE order\_no = coli.order\_no AND line\_item\_no = coli.line\_item\_no AND (coil\_status = 'CO' OR coil\_status = 'RD'))" + **WeightQuery** + "AND coli.mill\_width\_aim >= " + **WidthMinQuery** + " AND coli.mill\_width\_aim<=" + **WidthMaxQuery** + **ExcludeFillerBarQuery** + **MillExternalQuery** + "order by coli.mill\_external, coli.mill\_width\_aim, coli.min\_coil\_weight, coli.planned\_week ".
* Flush Failed coils:
  + Validate the width assessment (specified at the end of this process), if it is failed, then remove it from the list)
* Get the coil information for the matching coils:
  + When ViewUniqueCoilsOnly parameter is true:
    - SELECT coil.line\_item\_coil\_no, coil.mill\_weight\_aim, coil.coil\_status, coil.change\_datetime, customer.name, coli.promised\_date, coli.steel\_design\_cd, coli.caster\_internal,coli.caster\_external,coli.mill\_external,clip.price\_override\_reason\_cd FROM coil, customer\_order\_line\_item coli, customer, cust\_order\_line\_item\_price clip WHERE coil.order\_no =[COIL ORDER] AND coil.line\_item\_no = [COIL LINE ITEM] AND coil.grade = [COIL GRADE] AND coil.order\_no = coli.order\_no AND coil.line\_item\_no = coli.line\_item\_no AND customer.customer\_id = coli.customer\_id AND coil.order\_no = clip.order\_no AND coil.line\_item\_no = clip.line\_item\_no AND coil.line\_item\_coil\_no in ( SELECT MIN(line\_item\_coil\_no)FROM coil "WHERE order\_no = [COIL ORDER ]AND line\_item\_no = [COIL LINE ITEM] AND ( coil\_status = 'CO' OR coil\_status = 'RD' ) GROUP BY coil\_status )
  + When ViewUniqueCoilsOnly parameter is false:
* SELECT coil.line\_item\_coil\_no, coil.mill\_weight\_aim, coil.coil\_status, coil.change\_datetime, customer.name, coli.promised\_date, coli.steel\_design\_cd, coli.caster\_internal, coli.caster\_external, coli.mill\_external, clip.price\_override\_reason\_cd FROM coil, customer\_order\_line\_item coli, customer, cust\_order\_line\_item\_price clipWHERE coil.order\_no = [COIL ORDER] AND coil.line\_item\_no = [COIL LINE ITEM] AND coil.grade = [COIL GRADE] AND (coil.coil\_status = 'CO' OR coil.coil\_status = 'RD') AND coil.order\_no = coli.order\_no AND coil.line\_item\_no = coli.line\_item\_no AND coil.order\_no = clip.order\_no AND coil.line\_item\_no = clip.line\_item\_no AND customer.customer\_id = coli.customer\_id
* Separate the result of the query in two sets: one for the CO status slabs and other for the RD status slabs.
* Start refreshing screens (start timer)

**Width Assesment**

* Width Assessment (WEASSESS.CPP Line 31)
  + Obtain the Delta for the selected slab:
    - SELECT ROUND(max\_slab\_width\_delta,1), ROUND(min\_slab\_width\_delta,1) FROM sb\_std\_slab\_widths WHERE from\_width <= [SELECTED SLAB WIDTH] AND ([SELECTED SLAB WIDTH] < to\_width OR ([SELECTED SLAB WIDTH] <= to\_width and last\_row = 'Y'))"
    - Where [SELECTED SLAB WIDTH] is the width of the selected slab.
  + Calculate the slab cold coil minimum and maximum:
    - sColdCoilMin= MillWidthAim + MinDelta
    - sColdCoilMax=MillWidthAim + MaxDelta
  + Perform the following Validations:
    - if (!((fWidthAtHead \*100) >= (fMillWidthAim + fMinDelta)\*100))

{

return FALSE;

}

* + - if(!((fWidthAtHead \*100) <= (fMillWidthAim + fMaxDelta)\*100))

{

return FALSE;

}

* + - if(!((fWidthAtTail \*100) >= (fMillWidthAim + fMinDelta)\*100))

{

return FALSE;

}

* + - if(!((fWidthAtTail \*100) <= (fMillWidthAim + fMaxDelta)\*100))

{

return FALSE;

}

# UC064 – Slab Re-Application – Display Matching Required Disposition Coils for a Slab (Implemented on UC063)

## Description

This functionality displays the list of orders for the “Required Disposition” coils, that matches the criteria specified.

## Triggers

* When the actor enters into the Slab Re-Application Dashboard, and select the “Required Disposition” coils.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

# UC065 – Slab Re-Application – Display Matching Stock Coils for a Slab (Implemented on UC063)

## Description

This functionality displays the list of coils for the “Stock” tab, that matches the criteria specified.

## Triggers

* When the actor enters into the Slab Re-Application Dashboard, and select the “Stock” coils.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## End of use case

## Alternative and Exception Flows

## No alternate or exception workflows in this use case

## Business Rules

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical specification

# UC066 – Slab Re-Application – Re-Apply Slab

## Description

This functionality allows to reapply the slab to another coil so it can be assigned to another order. displays the list of coils for the “Stock” tab, that matches the criteria specified.

## Triggers

* When the actor enters into the Slab Re-Application Dashboard, and clicks on the “Re-Apply Slab” button.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects one of the slabs that are on the “Slab on Tunnel Furnace” List.

## The system executes UC065 and UC064 to display the possible orders to re-apply the slab

## The user selects the order from the “Committed Coils” or “Required Disposition” tabs (BR1) (AF1)

## The user clicks on the “Re-Apply Slab” button

## The system opens a modal window displaying the information of the current order, the new order, and the slab information, new order tolerances and operational problem to be selected by the user, with the following fields:

* Current Order For Slab
  + Customer Name
  + Order Number
  + Line Item
  + Gauge Aim
  + Promised Date
  + Steel Design Code
* New Order for Slab
  + Customer Name
  + Order Number
  + Line Item
  + Gauge Aim
  + Promised Date
  + Steel Design Code
* Produced Slab Information
  + Slab Number
  + Weight
  + Produce Date
  + Length
  + Head Width
  + Tail Width
* New Order Tolerances
  + Length Min
  + Length Max
  + Gauge Min
  + Gauge Max
  + Width Min
  + Width Max
  + Grade
* Operational Problem
  + Caster
  + Tunnel Furnace
  + Roughing Mill
  + Finishing Mill
  + Coiler
  + Other

## The user selects one of the Operational Problems.

## The user reviews the information displayed to verify that’s correct and clicks on the “Confirm” button.

## The system Re-Applies the slab to the selected order

## End of use case.

## Alternative and Exception Flows

## AF1 – Use Stock Coils

## The user selects the “Stock” tab

## The system executes UC066 to get the list of stock coils

## The flow continues to step 38.5.1.4.

## Business Rules

## BR1 – Only one order selected

## The Selection of only one row is allowed between the three tabs “Stock”, “Committed Coils” and “Requires Disposition”.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Current Order – Customer Name | Output | Yes | Value prefilled from the selected slab and order to reapply. |  | customer\_order\_line\_item | Customer\_id |
| Current Order – Order Number | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Order\_no |
| Current Order – Line Item | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Line\_item\_no |
| Current Order – Gauge Aim | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Mill\_gauge\_aim |
| Current Order – Promised Date | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Promised\_date |
| Current Order – Steel Design Code | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Steel\_design\_cd |
| New Order – Customer Name | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Customer\_id |
| New Order – Order Number | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Order\_no |
| New Order – Line Item | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Line\_item\_no |
| New Order – Gauge Aim | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Mill\_gauge\_aim |
| New Order – Promised Date | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Promised\_date |
| New Order – Steel Design Code | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Steel\_design\_cd |
| Produced Slab Information – Slab Number | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Slab\_no |
| Produced Slab Information – Weight | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Piece\_weight |
| Produced Slab Information – Produced Date | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Produced\_dt\_stamp |
| Produced Slab Information – Head Width | Output | Yes | Value prefilled from the selected slab and order to reapply | One decimal | caster\_thin\_slab\_report | Width\_at\_head |
| Produced Slab Information – Length | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Length\_cut |
| Produced Slab Information – Tail Width | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Width\_at\_tail |
| New Order Tolerances – Length Min | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | (min\_coil\_weight)/(min\_width\*102\*.00000785) |
| New Order Tolerances – Length Max | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | (max\_coil\_weight)/(max\_width\*102\*.00000785) |
| New Order Tolerances – Gauge Min | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Gauge\_min |
| New Order Tolerances – Gauge Max | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Gauge\_max |
| New Order Tolerances –Width Min | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Min\_width |
| New Order Tolerances – Width Max | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Max\_width |
| Operational Problem – Caster | Input | Optional | Selected | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem – Tunnel Furnace | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem – Roughing Mill | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem – Finishing Mill | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem – Coiler | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem - Other | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Technical Specification

* For Step 38.17.1.5 on the main flow, to obtain the slab detail the following queries must be executed (SR001VW.CPP):
  + Get the selected slab from the list that are still on the furnace.
  + Validate if there is something selected from the list, if not finish the execution.
  + Get the selected slab data.
  + Check which tab is selected, in order to determine the type of order:
    - Validate if there is something selected on the committed orders, if nothing is selected, finish the execution. If there is a selected coil, get the information.
    - Validate if there is something selected on the Order Disposition List, if nothing is selected, finish the execution. If there is a selected coil, get the information.
    - Validate if there is something selected on the stock orders, if nothing is selected, finish the execution. If there is a selected coil, get the information.
  + Opens the dialog with the required information.
  + Get the selected slab Customer.
  + If the customer of the selected slab is not “STOCK”:
    - Set the order number of the selected slab on the current order field
    - Set the line item number of the selected slab on the line item number field
    - Set the promised date of the selected slab on the promised date field
    - Set the gauge aim of the selected slab on the gauge aim field
    - Set the steel design of the selected slab on the steel design field
  + Execute the Reapply on the switch (CONFDLG.CPP line 271) to obtain the current order information:
    - SELECT steel\_design\_cd, promised\_date, customer\_id, min\_coil\_weight, max\_coil\_weight, ROUND(min\_width,1), ROUND(max\_width,1), ROUND(min\_gauge,1), ROUND(max\_gauge,1), ROUND(mill\_gauge\_aim, 1) FROM customer\_order\_line\_item WHERE order\_no = [CURRENT ORDER NO] AND line\_item\_no = [CURRENT ITEM NUMBER].
    - [CURRENT ITEM NUMBER] is the line item number for the slab to be reapplied
    - [CURRENT ORDER NO] is the order number for the slab to be reapplied
  + Calculate the min and max weight:
    - SELECT std\_slab\_width FROM sb\_std\_slab\_widths WHERE [MinWidth] >= from\_width AND [MinWidth] <= to\_width
    - MinWeight / sStdWidthMin \* 90 \* .00000785
    - SELECT std\_slab\_width FROM sb\_std\_slab\_widths WHERE [MaxWidth] >= from\_width AND [MaxWidth] <= to\_width
    - MaxWeight / sStdWidthMax \* 90 \* .00000785
  + Data to be shown on the dialog must be obtained from the objects selected.
* For step 36.17.1.8 make the Slab Re-Application (SR001Doc.cpp line 380):
  + Get the Steel Density = SELECT avg\_steel\_density FROM sb\_cas\_constraints
  + Validate if the order is planned
    - Get the order number, line item number, line item coil number and grade from the slab.
    - Validate if the slab to which is going to be reapplied is stock. If it is:
      * Create database transaction
      * Update the Slab List for Stock Order:
        + UPDATE mps\_slab\_list SET hsm\_schedule\_no = NULL, sched\_slab\_seq = NULL, sched\_heat\_seq = NULL, sched\_string\_seq = NULL, order\_no = NULL, line\_item\_no = NULL, line\_item\_coil\_no = NULL, stock\_id = [CURRENT STOCK ID], mill\_gauge\_aim = [CURRENT MILL GAUGE AIM], mill\_width\_aim = [CURRENT MILL WIDTH AIM], mill\_weight\_aim = [CURRENT MILL WEIGHT AIM], slab\_thickness = [CURRENT SLAB THICKNESS], slab\_width\_head = [CURRENT WIDTH HEAD], slab\_width\_tail = [CURRENT WIDTH TAIL], slab\_length = [CURRENT SLAB LENGTH], mill\_weight\_min = convert(int, (0.9 \* [CURRENT WEIGHT])), mill\_weight\_max = convert(int, (1.1 \* [CURRENT WEIGHT])), coil\_length = NULL WHERE mill\_prod\_slab\_seq = [ CURRENT MILL PRODUCTION SLAB SEQUENCE]
        + Where:

[CURRENT STOCK ID] is the stock id of the selected slab to be reapplied.

CURRENT MILL GAUGE AIM] ] is the gauge aim of the selected slab to be reapplied

[CURRENT MILL WIDTH AIM] ] is the width aim of the selected slab to be reapplied

[CURRENT MILL WEIGHT AIM] ] is the weight of the selected slab to be reapplied

[CURRENT SLAB THICKNESS] ] is the slab thickness of the selected slab to be reapplied

[CURRENT WIDTH HEAD] ] is the width head of the selected slab to be reapplied

[CURRENT WIDTH TAIL] ] is the width tail of the selected slab to be reapplied

[CURRENT SLAB LENGTH] ] is the slab length of the selected slab to be reapplied

[CURRENT WEIGHT] ] is the current weight of the selected slab to be reapplied

[ CURRENT MILL PRODUCTION SLAB SEQUENCE] is the current mill production slab sequence of the selected slab to be reapplied

* + - * Update the production comment :
        + UPDATE caster\_string\_slab SET production\_comments = '[COMMENTS]' WHERE hsm\_schedule\_no = [CURRENT SCHEDULE] AND sched\_slab\_seq =[CURRENT SLAB SCHEDULE SEQUENCE]
        + Where:

[CURRENT SCHEDULE] is the active schedule

[CURRENT SLAB SCHEDULE SEQUENCE] is the sequence for the slab that is being reapplied

* + - * Update the Coil Status:
        + UPDATE coil SET coil\_status = 'RD' WHERE order\_no =[CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM] AND line\_item\_coil\_no = [CURRENT LINE ITEM]
        + If the changeDateTime is not empty, then add this statement to the query: “AND change\_datetime=[CHANGE DATE TIME]
        + Where:

[CURRENT LINE ITEM] is the line item of the slab to be reapplied

[CURRENT ORDER] is the order of the slab to be reapplied

[CHANGE DATE TIME] is the date and time in which the update is performed

* + - * Update the Line Item Status:
        + UPDATE customer\_order\_line\_item SET order\_no = order\_no WHERE order\_no = [CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM]
        + Where:

[CURRENT ORDER] is the order for the slab to be reapplied

[CURRENT LINE ITEM] is the line item number of the slab to be reapplied

* + - * If any of the previous updates fails, an error is thrown and logged in.
      * After committing the transaction, a message to Level 2 to send the Coil Id is sent (UC030).
    - If it is not stock.
      * Update the Slab List with the new Customer
        + UPDATE mps\_slab\_list SET hsm\_schedule\_no = NULL , sched\_slab\_seq = 'NULL',

sched\_heat\_seq = 'NULL', sched\_string\_seq = 'NULL' , stock\_id = 'NULL'

, slab\_width\_std = 'NULL'

,order\_no = [CURRENT ORDER]

, line\_item\_no = [CURRENT LINE ITEM NUMBER]

, line\_item\_coil\_no = [LINE ITEM COIL NUMBER]

, mill\_width\_aim = [CURRENT WIDTH AIM]

, mill\_gauge\_aim = [CURRENT GAUGE AIM]

, mill\_weight\_aim = [CURRENT WEIGHT AIM]

, slab\_thickness = [CURRENT SLAB THICKNESS]

, slab\_width\_head = [CURRENT SLAB WIDTH HEAD]

, slab\_width\_tail = [CURRENT SLAB WIDTH TAIL]

, slab\_length = [CURRENT SLAB LENGTH]

, mill\_weight\_min = [CURRENT COIL MIN WEIGHT]

, mill\_weight\_max = [CURRENT COIL MAX WEIGHT]

, coil\_length =[CURRENT COIL LENGTH]

WHERE mill\_prod\_slab\_seq = [CURRENT SLAB SEQUENCE]

* + - * + Where:

[CURRENT ORDER] is the order of the coil to be reapplied

[CURRENT LINE ITEM NUMBER] is the current line item number of the coil to be reapplied

[LINE ITEM COIL NUMBER] is the coil number of the coil to be reapplied

[CURRENT WIDTH AIM] is the aim width of the coil to be reapplied

[CURRENT GAUGE AIM] is the gauge aim of the coil to be reapplied

[CURRENT WEIGHT AIM] is the weight aim of the coil to be reapplied

[CURRENT SLAB THICKNESS] is the current thickness of the coil to be reapplied

[CURRENT SLAB WIDTH HEAD] is the current slab width on the head of the coil to be reapplied

[CURRENT SLAB WIDTH TAIL] is the current slab wdth on the tail of the coil to be reapplied

[CURRENT SLAB LENGTH] is the slab length of the coil to be reapplied

[CURRENT COIL MIN WEIGHT] is the minimum weight of the coil to be reapplied

[CURRENT COIL MAX WEIGHT] is the maximum weight of the coil to be reapplied

[CURRENT COIL LENGTH] is the current coil length of the coil to be reapplied

[CURRENT SLAB SEQUENCE] is the current slab sequence number of the coil to be reapplied

* + - * Update the production comment :
        + UPDATE caster\_string\_slab SET production\_comments = '[COMMENTS]' WHERE hsm\_schedule\_no = [CURRENT SCHEDULE] AND sched\_slab\_seq =[CURRENT SLAB SCHEDULE SEQUENCE]
        + Where:

[CURRENT SCHEDULE] is the active schedule

[CURRENT SLAB SCHEDULE SEQUENCE] is the sequence for the slab that is being reapplied

* + - * Update the Coil Status:
        + UPDATE coil SET coil\_status = 'RA' WHERE order\_no =[CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM] AND line\_item\_coil\_no = [CURRENT LINE ITEM]
        + If the changeDateTime is not empty, then add this statement to the query: “AND change\_datetime=[CHANGE DATE TIME]
        + Where:

[CURRENT LINE ITEM] is the line item of the slab to be reapplied

[CURRENT ORDER] is the order of the slab to be reapplied

[CHANGE DATE TIME] is the change date time of the slab to be reapplied

* + - * Validate a grade mismatch:
        + If the grade of the coil is different to the grade of the slab:

update coil\_process\_data SET ra\_grade\_mismatch = 'Y' where slab\_no ='[CURRENT SLAB]' and l3\_coil\_id = '[CURRENT COIL]'

* + - * + Where:

[CURRENT SLAB] is the slab to be reapplied

[CURRENT COIL] is the coil to which the slab is going to be reapplied

* + - * Update the Coil Status:
        + UPDATE coil SET coil\_status = 'RD' WHERE order\_no =[CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM] AND line\_item\_coil\_no = [CURRENT LINE ITEM]
        + If the changeDateTime is not empty, then add this statement to the query: “AND change\_datetime=[CHANGE DATE TIME]
        + Where:

[CURRENT LINE ITEM] is the line item of the slab to be reapplied

[CURRENT ORDER] is the order of the slab to be reapplied

[CHANGE DATE TIME] is the date and time in which the update is performed

* + - * Update the Line Item Status:
        + UPDATE customer\_order\_line\_item SET order\_no = order\_no WHERE order\_no = [CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM]
        + Where:

[CURRENT ORDER] is the order for the slab to be reapplied

[CURRENT LINE ITEM] is the line item number of the slab to be reapplied

* + - * If any of the previous updates fails, an error is thrown and logged in.
      * After committing the transaction, a message to Level 2 to send the Coil Id is sent (UC030).
  + If not planned, Validate if the slab is in a planned stock order
    - Validate if the slab is going to be applied to a stock order. If it does:
      * Update the slab list for the stock order:
        + Begin a database transaction
        + UPDATE mps\_slab\_list SET hsm\_schedule\_no = NULL, sched\_slab\_seq = NULL, sched\_heat\_seq = NULL, sched\_string\_seq = NULL, order\_no = NULL, line\_item\_no = NULL, line\_item\_coil\_no = NULL, stock\_id = [CURRENT STOCK ID], mill\_gauge\_aim = [CURRENT MILL GAUGE AIM], mill\_width\_aim = [CURRENT MILL WIDTH AIM], mill\_weight\_aim = [CURRENT MILL WEIGHT AIM], slab\_thickness = [CURRENT SLAB THICKNESS], slab\_width\_head = [CURRENT WIDTH HEAD], slab\_width\_tail = [CURRENT WIDTH TAIL], slab\_length = [CURRENT SLAB LENGTH], mill\_weight\_min = convert(int, (0.9 \* [CURRENT WEIGHT])), mill\_weight\_max = convert(int, (1.1 \* [CURRENT WEIGHT])), coil\_length = NULL WHERE mill\_prod\_slab\_seq = [ CURRENT MILL PRODUCTION SLAB SEQUENCE]
        + Where:

[CURRENT STOCK ID] is the stock id of the selected slab to be reapplied.

CURRENT MILL GAUGE AIM] ] is the gauge aim of the selected slab to be reapplied

[CURRENT MILL WIDTH AIM] ] is the width aim of the selected slab to be reapplied

[CURRENT MILL WEIGHT AIM] ] is the weight of the selected slab to be reapplied

[CURRENT SLAB THICKNESS] ] is the slab thickness of the selected slab to be reapplied

[CURRENT WIDTH HEAD] ] is the width head of the selected slab to be reapplied

[CURRENT WIDTH TAIL] ] is the width tail of the selected slab to be reapplied

[CURRENT SLAB LENGTH] ] is the slab length of the selected slab to be reapplied

[CURRENT WEIGHT] ] is the current weight of the selected slab to be reapplied

[ CURRENT MILL PRODUCTION SLAB SEQUENCE] is the current mill production slab sequence of the selected slab to be reapplied

* + - * + Commit the transaction, if it fails an error message must be logged.
      * If the order where is going to be reapplied is not stock, then:
        + Update the slab list for the new order:

UPDATE mps\_slab\_list SET hsm\_schedule\_no = NULL , sched\_slab\_seq = 'NULL',

sched\_heat\_seq = 'NULL', sched\_string\_seq = 'NULL' , stock\_id = 'NULL'

, slab\_width\_std = 'NULL'

,order\_no = [CURRENT ORDER]

, line\_item\_no = [CURRENT LINE ITEM NUMBER]

, line\_item\_coil\_no = [LINE ITEM COIL NUMBER]

, mill\_width\_aim = [CURRENT WIDTH AIM]

, mill\_gauge\_aim = [CURRENT GAUGE AIM]

, mill\_weight\_aim = [CURRENT WEIGHT AIM]

, slab\_thickness = [CURRENT SLAB THICKNESS]

, slab\_width\_head = [CURRENT SLAB WIDTH HEAD]

, slab\_width\_tail = [CURRENT SLAB WIDTH TAIL]

, slab\_length = [CURRENT SLAB LENGTH]

, mill\_weight\_min = [CURRENT COIL MIN WEIGHT]

, mill\_weight\_max = [CURRENT COIL MAX WEIGHT]

, coil\_length =[CURRENT COIL LENGTH]

WHERE mill\_prod\_slab\_seq = [CURRENT SLAB SEQUENCE]

* + - * + Where:

[CURRENT ORDER] is the order of the coil to be reapplied

[CURRENT LINE ITEM NUMBER] is the current line item number of the coil to be reapplied

[LINE ITEM COIL NUMBER] is the coil number of the coil to be reapplied

[CURRENT WIDTH AIM] is the aim width of the coil to be reapplied

[CURRENT GAUGE AIM] is the gauge aim of the coil to be reapplied

[CURRENT WEIGHT AIM] is the weight aim of the coil to be reapplied

[CURRENT SLAB THICKNESS] is the current thickness of the coil to be reapplied

[CURRENT SLAB WIDTH HEAD] is the current slab width on the head of the coil to be reapplied

[CURRENT SLAB WIDTH TAIL] is the current slab wdth on the tail of the coil to be reapplied

[CURRENT SLAB LENGTH] is the slab length of the coil to be reapplied

[CURRENT COIL MIN WEIGHT] is the minimum weight of the coil to be reapplied

[CURRENT COIL MAX WEIGHT] is the maximum weight of the coil to be reapplied

[CURRENT COIL LENGTH] is the current coil length of the coil to be reapplied

[CURRENT SLAB SEQUENCE] is the current slab sequence number of the coil to be reapplied

* + - * Update the Coil Status:
        + UPDATE coil SET coil\_status = 'RA' WHERE order\_no =[CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM] AND line\_item\_coil\_no = [CURRENT LINE ITEM]
        + If the changeDateTime is not empty, then add this statement to the query: “AND change\_datetime=[CHANGE DATE TIME]
        + Where:

[CURRENT LINE ITEM] is the line item of the slab to be reapplied

[CURRENT ORDER] is the order of the slab to be reapplied

[CHANGE DATE TIME] is the date and time in which the update is performed

* + - * After committing the transaction, a message to Level 2 to send the Coil Id is sent (UC030).
  + If not planned nor stock then is unplanned
    - * Validate if the order to which is going to be applied is stock. If it is:
        + Begin a database transaction
        + UPDATE mps\_slab\_list SET hsm\_schedule\_no = NULL, sched\_slab\_seq = NULL, sched\_heat\_seq = NULL, sched\_string\_seq = NULL, order\_no = NULL, line\_item\_no = NULL, line\_item\_coil\_no = NULL, stock\_id = [CURRENT STOCK ID], mill\_gauge\_aim = [CURRENT MILL GAUGE AIM], mill\_width\_aim = [CURRENT MILL WIDTH AIM], mill\_weight\_aim = [CURRENT MILL WEIGHT AIM], slab\_thickness = [CURRENT SLAB THICKNESS], slab\_width\_head = [CURRENT WIDTH HEAD], slab\_width\_tail = [CURRENT WIDTH TAIL], slab\_length = [CURRENT SLAB LENGTH], mill\_weight\_min = convert(int, (0.9 \* [CURRENT WEIGHT])), mill\_weight\_max = convert(int, (1.1 \* [CURRENT WEIGHT])), coil\_length = NULL WHERE mill\_prod\_slab\_seq = [ CURRENT MILL PRODUCTION SLAB SEQUENCE]
        + Where:

[CURRENT STOCK ID] is the stock id of the selected slab to be reapplied.

CURRENT MILL GAUGE AIM] ] is the gauge aim of the selected slab to be reapplied

[CURRENT MILL WIDTH AIM] ] is the width aim of the selected slab to be reapplied

[CURRENT MILL WEIGHT AIM] ] is the weight of the selected slab to be reapplied

[CURRENT SLAB THICKNESS] ] is the slab thickness of the selected slab to be reapplied

[CURRENT WIDTH HEAD] ] is the width head of the selected slab to be reapplied

[CURRENT WIDTH TAIL] ] is the width tail of the selected slab to be reapplied

[CURRENT SLAB LENGTH] ] is the slab length of the selected slab to be reapplied

[CURRENT WEIGHT] ] is the current weight of the selected slab to be reapplied

[ CURRENT MILL PRODUCTION SLAB SEQUENCE] is the current mill production slab sequence of the selected slab to be reapplied

* + - * After committing the transaction, a message to Level 2 to send the Coil Id is sent (UC030).
      * Else, if it is not a stock order, then it’s a planned order.
        + UPDATE mps\_slab\_list SET hsm\_schedule\_no = NULL , sched\_slab\_seq = 'NULL',

sched\_heat\_seq = 'NULL', sched\_string\_seq = 'NULL' , stock\_id = 'NULL'

, slab\_width\_std = 'NULL'

,order\_no = [CURRENT ORDER]

, line\_item\_no = [CURRENT LINE ITEM NUMBER]

, line\_item\_coil\_no = [LINE ITEM COIL NUMBER]

, mill\_width\_aim = [CURRENT WIDTH AIM]

, mill\_gauge\_aim = [CURRENT GAUGE AIM]

, mill\_weight\_aim = [CURRENT WEIGHT AIM]

, slab\_thickness = [CURRENT SLAB THICKNESS]

, slab\_width\_head = [CURRENT SLAB WIDTH HEAD]

, slab\_width\_tail = [CURRENT SLAB WIDTH TAIL]

, slab\_length = [CURRENT SLAB LENGTH]

, mill\_weight\_min = [CURRENT COIL MIN WEIGHT]

, mill\_weight\_max = [CURRENT COIL MAX WEIGHT]

, coil\_length =[CURRENT COIL LENGTH]

WHERE mill\_prod\_slab\_seq = [CURRENT SLAB SEQUENCE]

* + - * + Where:

[CURRENT ORDER] is the order of the coil to be reapplied

[CURRENT LINE ITEM NUMBER] is the current line item number of the coil to be reapplied

[LINE ITEM COIL NUMBER] is the coil number of the coil to be reapplied

[CURRENT WIDTH AIM] is the aim width of the coil to be reapplied

[CURRENT GAUGE AIM] is the gauge aim of the coil to be reapplied

[CURRENT WEIGHT AIM] is the weight aim of the coil to be reapplied

[CURRENT SLAB THICKNESS] is the current thickness of the coil to be reapplied

[CURRENT SLAB WIDTH HEAD] is the current slab width on the head of the coil to be reapplied

[CURRENT SLAB WIDTH TAIL] is the current slab wdth on the tail of the coil to be reapplied

[CURRENT SLAB LENGTH] is the slab length of the coil to be reapplied

[CURRENT COIL MIN WEIGHT] is the minimum weight of the coil to be reapplied

[CURRENT COIL MAX WEIGHT] is the maximum weight of the coil to be reapplied

[CURRENT COIL LENGTH] is the current coil length of the coil to be reapplied

[CURRENT SLAB SEQUENCE] is the current slab sequence number of the coil to be reapplied

* + - * Update the production comment :
        + UPDATE caster\_string\_slab SET production\_comments = '[COMMENTS]' WHERE hsm\_schedule\_no = [CURRENT SCHEDULE] AND sched\_slab\_seq =[CURRENT SLAB SCHEDULE SEQUENCE]
        + Where:

[CURRENT SCHEDULE] is the active schedule

[CURRENT SLAB SCHEDULE SEQUENCE] is the sequence for the slab that is being reapplied

* + - * Update the Coil Status:
        + UPDATE coil SET coil\_status = 'RA' WHERE order\_no =[CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM] AND line\_item\_coil\_no = [CURRENT LINE ITEM]
        + If the changeDateTime is not empty, then add this statement to the query: “AND change\_datetime=[CHANGE DATE TIME]
        + Where:

[CURRENT LINE ITEM] is the line item of the slab to be reapplied

[CURRENT ORDER] is the order of the slab to be reapplied

* + - * + [CHANGE DATE TIME] is the change date time of the slab to be reapplied
      * After committing the transaction, a message to Level 2 to send the Coil Id is sent (UC030).

# UC067 – Slab Re-Application – Change search criteria for matching orders

## Description

This functionality allows to change the criteria for search the matching orders available to re-apply a slab.

## Triggers

* When the actor enters into the Slab Re-Application Dashboard, and select the “Properties” button.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Properties” tab.

## The system displays a modal window with the following information prefilled with the current values on the system:

## Mill External Rating at or Above

## Width Above

## Width Below

## Include Filler Bars in Search

## Search only by this grade.

## The user enters the required information (BR1)

## The user clicks on the “Save” button and stores the properties for the order search (AF1).

## End of use case

## Alternative and Exception Flows

## AF1 – Cancel Button

## The user clicks on the “Cancel” button

## The system closes the modal window

## End of use case.

## Business Rules

## BR1 – Search Criteria Conditions

## The data will require de following validations:

## Mill External Rating at or Above: filters by quality rating. Higher rating means that more quality is required for the order, so orders with higher quality rating will not show up.

## Width constraints Above/Below: find orders with bigger or smaller widths

## Include filler bars in search: checking this box allows to get filler orders

## Search only by this grade: force the system to search for orders of the selected grade.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Mill External Rating at Or Above | Input/Output | No | What is populated from the system | N/A | Nb\_reapp\_parameters | Mill\_External\_Rating |
| Width Constraint Above | Input/Output | No | What is populated from the system | N/A | Nb\_reapp\_parameters | Width\_Above |
| Width Constraint Below | Input/Output | No | What is populated from the system | N/A | Nb\_reapp\_parameters | Width\_Below |
| Include Filler Bars In Search | Input/Output | No | What is populated from the system | N/A | Nb\_reapp\_parameters | Filler\_bars |
| Search Only This Grade | Input/Output | No | What is populated from the system | N/A | Nb\_reapp\_parameters | Grade\_flag |
| Search Grade | Input/Output | No | What is populated from the system | N/A | Nb\_reapp\_parameters | Manual\_grade |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to perform step 37.5.1.2 on the main flow, this query must be executed (SQL.CPP Line 4144):
  + SELECT mill\_external\_rating, width\_above, width\_below, filler\_bars, manual\_grade, grade\_flag, weight\_min\_string, weight\_max\_string, ROUND(weight\_min\_percent, 3), ROUND(weight\_max\_percent, 3) FROM nb\_reapp\_parameters WHERE id\_no = 1
* In order to perform step 37.1.4 on the main flow, this query must be executed(SQL.CPP Line 4214):
  + UPDATE nb\_reapp\_parameters SET mill\_external\_rating = [NEW MILL EXT RATING] , width\_above = [NEW WIDTH ABOVE], width\_below = [NEW WIDTH BELOW], filler\_bars = [NEW FILLER BARS], manual\_grade = [NEW MANUAL GRADE], grade\_flag = [NEW GRADE] WHERE id\_no = 1
  + Where:
    - [NEW MILL EXT RATING] is the mill rating value on the input field
    - [NEW WIDTH ABOVE] is the width above value on the input field
    - [NEW WIDTH BELOW] is the width below value on the input field
    - [NEW FILLER BARS] is the checkbox value on the filler bars field. If checked is true, if not is false.
    - [NEW MANUAL GRADE] is the checkbox value on the manual grade field. If checked is true, if not is false
    - [NEW GRADE ] is the value on the new grade combo box.

# UC068 – Slab Re-Application – Display Slab Details

## Description

This functionality allow to the user to see the slab detailed information.

## Triggers

* When the double-clicks on the slab from which wants to know the detail.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor double clicks on the slab from which wants to know the detail from the “Slabs in the Tunnel Furnace” list.

## The system displays a modal window with the following information from the slab (BR1):

## Slab Information

## L3 Coil Id

## Order Number

## Line Item Number

## Grade Aim

## Grade Family

## End Use

## Customer Name

## Promised Date

## Produced Date/Time

## Steel Design Code

## Gauge Aim

## Failed Assessments

## Width Assessments

## Weight Assessments

## Width

## Aim Coil Widths

## Cold Coil

## Hot Slab Head

## Hot Slab Tail

## Min Coil Width

## Max Coil Width

## Actual Slab Widths

## Hot Slab Head

## Hot Slab Tail

## Chemistry

## 

## Weight

## End of use case

## Alternative and Exception Flows

## There are no alternate or exception flows

## Business Rules

## BR1 – Read Only Fields

## All data displayed is read only.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| L3 Coil Id | Output | Yes | N/A | Grouped inside “Slab Information” area | Mps\_slab\_list | L3\_coil\_id |
| Customer Name | Output | Yes | N/A | Grouped inside “Slab Information” area | customer | Name |
| Order Number | Output | Yes | N/A | Grouped inside “Slab Information” area | Mps\_slab\_list | Order\_no |
| Promised Date | Output | Yes | N/A | Grouped inside “Slab Information” area | customer\_order\_line\_item | Promised\_date |
| Line Item Number | Output | Yes | N/A | Grouped inside “Slab Information” area | coil | Line\_item\_no |
| Produced Date Time | Output | Yes | N/A | Grouped inside “Slab Information” area | caster\_thin\_slab\_report | Produced\_dt\_stamp |
| Grade Aim | Output | Yes | N/A | Grouped inside “Slab Information” area | Mps\_slab\_list | Mill\_grade\_aim |
| Steel Design Code | Output | Yes | N/A | Grouped inside “Slab Information” area | customer\_order\_line\_item | Steel\_design\_cd |
| Grade Family | Output | Yes | N/A | Grouped inside “Slab Information” area | grade | Grade\_family\_cd |
| Gauge Aim | Output | Yes | N/A | Grouped inside “Slab Information” area | Mps\_slab\_list | Mill\_gauge\_aim |
| End Use | Output | Yes | N/A | Grouped inside “Slab Information” area | customer\_order\_line\_item | End\_use |
| Width Assessment | Output | Yes | N/A | Grouped by “Failed Assessment” area. Must be a checkbox | Result from Width Assessment |  |
| Weight Assessment | Output | Yes | N/A | Grouped by “Failed Assessment” area. Must be a checkbox | Result from WeightAssessment |  |
| Aim Coil Width – Cold Coil | Output | Yes | N/A | Inside the “Width” tab, group by “Aim Coil Width Group” | Mps\_slab\_list | mill\_width\_aim |
| Aim Coil Width – Hot Slab Head | Output | Yes | N/A | Inside the “Width” tab, group by “Aim Coil Width Group” | caster\_thin\_slab\_report | Width\_at\_head |
| Aim Coil Width – Hot Slab Tail | Output | Yes | N/A | Inside the “Width” tab, group by “Aim Coil Width Group” | caster\_thin\_slab\_report | Width\_at\_tail |
| Aim Coil Width – Min Coil Width | Output | Yes | N/A | Inside the “Width” tab, group by “Aim Coil Width Group” | Customer\_order\_line\_item | Min\_width |
| Aim Coil Width – Max Coil Width | Output | Yes | N/A | Inside the “Width” tab, group by “Aim Coil Width Group” | Customer\_order\_line\_item | Max\_width |
| Actual Slab Widths – Hot Slab Head | Output | Yes | N/A | Inside the “Width” tab, group by “Actual Slab Widths” | Mps\_slab\_list | Slab\_width\_head |
| Actual Slab Widths – Hot Slab Tail | Output | Yes | N/A | Inside the “Width” tab, group by “Actual Slab Widths” | Mps\_slab\_list | Slab\_width\_tail |
| Aim Coil Weight – Mill Aim Weight | Output | Yes | N/A | Inside the “Weight” tab group by “Aim coil weight” | Mps\_slab\_list | Mill\_weight\_aim |
| Aim Coil Weight – Min Coil Weight | Output | Yes | N/A | Inside the “Weight” tab group by “Aim coil weight” | Mps\_slab\_list | Mill\_weight\_min |
| Aim Coil Weight – Max Coil Weight | Output | Yes | N/A | Inside the “Weight” tab group by “Aim coil weight” | Mps\_slab\_list | Mill\_weight\_max |
| Actual Weight – Slab Weight | Output | Yes | N/A | Inside the “Weight” tab group by “Actual Weight” | Caster\_thin\_slab\_report | Piece\_weight |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

In order to perform step 38.5.1.2 on the main flow, the chemical, width and weight assessments, the following logic must be implemented:

* Check which tab is selected, in order to determine the type of order:
  + Validate if there is something selected on the committed orders, if nothing is selected, finish the execution. If there is a selected coil, get the information.
  + Validate if there is something selected on the Order Disposition List, if nothing is selected, finish the execution. If there is a selected coil, get the information.
  + Validate if there is something selected on the stock orders, if nothing is selected, finish the execution. If there is a selected coil, get the information.
* Opens the dialog with the slab information, which is specified on the tables and fields above. Weight and width assessments are specified below.

**Weight Assessment (WEASSESS.CPP Line 39)**

* + Obtain the maximum, minimum mill weight, and the slabweight.
  + Obtain the Yield loss.
  + Validate if the slab weight is less than the millweightmax +100
    - If it is, validate if the slabweight – slabweight\*(yieldloss/100) greater than or equal than the minimum mill weight
      * Return true
  + If both validations fails, then return false.

**Width Assessment (WASSESS.CPP Line 31)**

* + Obtain the Delta for the selected slab:
    - SELECT ROUND(max\_slab\_width\_delta,1), ROUND(min\_slab\_width\_delta,1) FROM sb\_std\_slab\_widths WHERE from\_width <= [SELECTED SLAB WIDTH] AND ([SELECTED SLAB WIDTH] < to\_width OR ([SELECTED SLAB WIDTH] <= to\_width and last\_row = 'Y'))"
    - Where [SELECTED SLAB WIDTH] is the width of the selected slab.
  + Calculate the slab cold coil minimum and maximum:
    - sColdCoilMin= MillWidthAim + MinDelta
    - sColdCoilMax=MillWidthAim + MaxDelta
  + Perform the following Validations:
    - if (!((fWidthAtHead \*100) >= (fMillWidthAim + fMinDelta)\*100))

{

return FALSE;

}

* + - if(!((fWidthAtHead \*100) <= (fMillWidthAim + fMaxDelta)\*100))

{

return FALSE;

}

* + - if(!((fWidthAtTail \*100) >= (fMillWidthAim + fMinDelta)\*100))

{

return FALSE;

}

* + - if(!((fWidthAtTail \*100) <= (fMillWidthAim + fMaxDelta)\*100))

{

return FALSE;

}

* + - If all past validations fails then return TRUE

# UC069 – Slab Re-Application – Display Order Details

## Description

This functionality allow to the user to see the order details.

## Triggers

* When the double-clicks on the order from which wants to know the detail from the “Committed Coils”, “Requires Disposition” lists.
* Selects an order from the “Committed Coils”, “Requires Disposition” tabs, and click on the Coil Details icon.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor double clicks on the order, or select it and click on the “Coil Detail” button, from which wants to know the detail from the “Slabs in the Tunnel Furnace” list.

## The system displays a modal window with the following information from the order (BR1):

## Customer Name

## Order Number

## Line Item

## Ordered Coil Weight

## Min Coil Weight

## Max Coil Weight

## Coil Temperature

## Finish Temperature

## Steel Design Code

## Promised Date

## Mill Width Aim

## Mill Width Min

## Mill Width Max

## Mill Gauge Aim

## Slab Length

## End Use

## End of use case

## Alternative and Exception Flows

## There are no alternate or exception flows

## Business Rules

## BR1 – Read Only Fields

## All data displayed is read only.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Customer Name | Output | N/A | N/A | N/A | Customer | Name |
| Order Number | Output | N/A | N/A | N/A | Mps\_slab\_list | Order\_no |
| Line Item | Output | N/A | N/A | N/A | Mps\_slab\_list | Line\_item\_no |
| Ordered Coil Weight | Output | N/A | N/A | N/A | Mps\_slab\_list | mill\_weight\_aim |
| Min Coil Weight | Output | N/A | N/A | N/A | Mps\_slab\_list | Mill\_weight\_min |
| Max Coil Weight | Output | N/A | N/A | N/A | Mps\_slab\_list | Mill\_weight\_max |
| Coil Temperature | Output | N/A | N/A | N/A | Rolling\_mill\_practice | Coiling\_temp\_aim |
| Finish Temperature | Output | N/A | N/A | N/A | Metallurgy\_steel\_design | Finish\_temp\_aim |
| Steel Design Code | Output | N/A | N/A | N/A | Customer\_order\_line\_item | Steel\_design\_cd |
| Promised Date | Output | N/A | N/A | N/A | Customer\_order\_line\_item | Promised\_date |
| Mill Width Aim | Output | N/A | N/A | N/A | Customer\_order\_line\_item | Mill\_width\_aim |
| Mill Width Min | Output | N/A | N/A | N/A | Customer\_order\_line\_item | Min\_width |
| Mill Width Max | Output | N/A | N/A | N/A | Customer\_order\_line\_item | Max\_width |
| Mill Gauge Aim | Output | N/A | N/A | N/A | Customer\_order\_line\_item | Mill\_gauge\_aim |
| Slab Length | Output | N/A | N/A | N/A | Mps\_slab\_list | Slab\_length |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

In order to display the order details, the following logic must be executed (SR001VW.CPP Line 1007):

* Check which tab is selected, in order to determine the type of order:
  + Validate if there is something selected on the committed orders, if nothing is selected, finish the execution. If there is a selected coil, get the information.
  + Validate if there is something selected on the Order Disposition List, if nothing is selected, finish the execution. If there is a selected coil, get the information.
  + Validate if there is something selected on the stock orders, if nothing is selected, finish the execution. If there is a selected coil, get the information.

Opens the dialog with the order information, which is obtained as follows:

* Get Coil Information:
  + SELECT promised\_date, steel\_design\_cd, customer\_id, boundary\_chemistry\_id, min\_coil\_weight, max\_coil\_weight, min\_width, max\_width FROM customer\_order\_line\_item WHERE order\_no = [SELECTED COIL ORDER] AND line\_item\_no = [SELECTED COIL LINE ITEM]
* Get Mill Width:
  + SELECT ROUND(min\_width,0), ROUND(max\_width,0) FROM customer\_order\_line\_item WHERE order\_no = [SELECTED COIL ORDER] AND line\_item\_no =[SELECTED COIL LINE ITEM NO]
* Get Rolling Mill Practices
  + SELECT coiling\_temp\_aim, finish\_temp\_aim FROM rolling\_mill\_practice rmp, metallurgy\_steel\_design msd WHERE msd.steel\_design\_cd = [SELECTED COIL STEEL DESIGN CD] AND msd.rolling\_mill\_practice\_cd = rmp.rolling\_mill\_practice\_cd
* Get End Use:
  + SELECT end\_use FROM customer\_order\_line\_item WHERE order\_no =[SELECTED COIL ORDER NO] AND line\_item\_no = [SELECTED COIL LINE ITEM NO]
* Any other information must be obtained from the detail of the selected coil.

# UC070 – Slab Re-Application – Retransmit PDI to rolling mill

## Description

This functionality allow to the user Transfer product data input (PDI) information to Level 2 Process Control systems.

## Triggers

* When the user clicks on the Re-Transmit PDI button.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the slab that he wants to retransmit

## The actor clicks on the “Retransmit PDI” button

## The system retransmit the PDI to L2 system (BR1).

## End of use case

## Alternative and Exception Flows

## There are no alternate or exception flows

## Business Rules

## BR1 – Send Message to L2

## Message from UC030 is sent.

## BR2 – Change Coil status

## Status for this coil is set to “RA”.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

Send message L3\_MPS\_COIL\_ADDED (UC029)

# UC071 - Slab Re-Application – Undo a slab re-application

## Description

This functionality allow to the user to revert the re-application of a slab to its original order. This will cause to send the PDI to level 2.

## Triggers

* When the user clicks on the “Revert” button.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the slab that he wants to revert.

## The actor clicks on the “Revert” button (FA1) (BR1)

## The system Displays the confirmation window displaying the following information

* Current Order For Slab
  + Customer Name
  + Order Number
  + Line Item
  + Gauge Aim
  + Promised Date
  + Steel Design Code
* New Order for Slab
  + Customer Name
  + Order Number
  + Line Item
  + Gauge Aim
  + Promised Date
  + Steel Design Code
* Produced Slab Information
  + Slab Number
  + Weight
  + Produce Date
  + Length
  + Head Width
  + Tail Width
* New Order Tolerances
  + Length Min
  + Length Max
  + Gauge Min
  + Gauge Max
  + Width Min
  + Width Max
  + Grade
* Operational Problem
  + Caster
  + Tunnel Furnace
  + Roughing Mill
  + Finishing Mill
  + Coiler
  + Other

## The user clicks on the “Confirm” button.

## The system reverts the slab to its original order (BR2) (BR3).

## End of use case

## Alternative and Exception Flows

## FA1 – Slab with status different than “RA”

## The system does not enable the revert button, since the selected slab is not on “RA” status.

## End use case.

## Business Rules

## BR1 – Slab that has been re-applied

## Only slabs that have been re-applied will be able to be reverted. Must have “RA” status.

## BR2 – Send Message to L2

## Message from UC030 is sent.

## BR3 – Change coil status

## Coil status has to change to “CO”.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Current Order – Customer Name | Output | Yes | Value prefilled from the selected slab and order to reapply. |  | customer\_order\_line\_item | Customer\_id |
| Current Order – Order Number | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Order\_no |
| Current Order – Line Item | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Line\_item\_no |
| Current Order – Gauge Aim | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Mill\_gauge\_aim |
| Current Order – Promised Date | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Promised\_date |
| Current Order – Steel Design Code | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Steel\_design\_cd |
| New Order – Customer Name | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Customer\_id |
| New Order – Order Number | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Order\_no |
| New Order – Line Item | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Line\_item\_no |
| New Order – Gauge Aim | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Mill\_gauge\_aim |
| New Order – Promised Date | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Promised\_date |
| New Order – Steel Design Code | Output | Yes | Value prefilled from the selected slab and order to reapply | From the new order selected on the matching orders | customer\_order\_line\_item | Steel\_design\_cd |
| Produced Slab Information – Slab Number | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Slab\_no |
| Produced Slab Information – Weight | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Piece\_weight |
| Produced Slab Information – Produced Date | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Produced\_dt\_stamp |
| Produced Slab Information – Head Width | Output | Yes | Value prefilled from the selected slab and order to reapply | One decimal | caster\_thin\_slab\_report | Width\_at\_head |
| Produced Slab Information – Length | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Length\_cut |
| Produced Slab Information – Tail Width | Output | Yes | Value prefilled from the selected slab and order to reapply |  | caster\_thin\_slab\_report | Width\_at\_tail |
| New Order Tolerances – Length Min | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | (min\_coil\_weight)/(min\_width\*102\*.00000785) |
| New Order Tolerances – Length Max | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | (max\_coil\_weight)/(max\_width\*102\*.00000785) |
| New Order Tolerances – Gauge Min | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Gauge\_min |
| New Order Tolerances – Gauge Max | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Gauge\_max |
| New Order Tolerances –Width Min | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Min\_width |
| New Order Tolerances – Width Max | Output | Yes | Value prefilled from the selected slab and order to reapply |  | customer\_order\_line\_item | Max\_width |
| Operational Problem – Caster | Input | Optional | Selected | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem – Tunnel Furnace | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem – Roughing Mill | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem – Finishing Mill | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem – Coiler | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |
| Operational Problem - Other | Input | Optional | Empty | It is a radio button. Only one of the fields from the Operational Problems can be selected. |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to revert the slab reapplication on step 45.5.1.5, the following activities must be performed on the system (SR001VW.CPP Line 708):
  + Stop Refreshing the screen
  + Validate if the selected slab is on the list. If it is:
    - Get the comment from the selected radio option.
    - Get item, line and order for the original status.
      * SELECT order\_no, line\_item\_no, line\_item\_coil\_no FROM caster\_string\_slab WHERE hsm\_schedule\_no =[ORIGINAL SCHEDULE] AND sched\_slab\_seq =[ORIGINAL SLAB SEQUENCE]
      * Where:
        + [ORIGINAL SCHEDULE] is the schedule where the slab was originally assigned (before the reapplication)
        + [ORIGINAL SLAB SEQUENCE] is the sequence that the slab had originally assigned (before the reapplication)
      * If the result is greater than 0 ( that means that the original order and slab where found), then:
        + Get the caster information for the slab:

SELECT order\_no, line\_item\_no,grade, ROUND(mill\_gauge\_aim,2), ROUND(mill\_width\_aim,1), ROUND(slab\_width\_head,1),ROUND(slab\_width\_tail,1),mill\_weight\_aim, sched\_heat\_seq, sched\_string\_seq, line\_item\_coil\_no, stock\_id, slab\_thickness, slab\_length, coil\_length, casting\_time, casting\_speed, planned\_slab\_type, slab\_width\_change\_mode FROM caster\_string\_slab WHERE hsm\_schedule\_no = [ORIGINAL SCHEDULE] AND sched\_slab\_seq = [ORIGINAL SLAB SEQUENCE]

Where:

[ORIGINAL SCHEDULE] is the schedule where the slab was originally assigned (before the reapplication)

[ORIGINAL SLAB SEQUENCE] is the sequence that the slab had originally assigned (before the reapplication)

* + - * + Get the Caster String Slab:

SELECT order\_no, line\_item\_no, grade, ROUND(mill\_gauge\_aim,2), ROUND(mill\_width\_aim,1), ROUND(slab\_width\_head,1), ROUND(slab\_width\_tail,1), mill\_weight\_aim, sched\_heat\_seq, sched\_string\_seq, line\_item\_coil\_no, stock\_id, slab\_thickness, slab\_length, coil\_length, casting\_time, casting\_speed, planned\_slab\_type, slab\_width\_change\_mode FROM caster\_string\_slab WHERE hsm\_schedule\_no = [SCHEDULE NUMBER FROM SLAB] AND sched\_slab\_seq = [SELECTED SLAB SEQUENCE]

* + - * + Get the Coil Status for the coil:

SELECT coil\_status FROM coil WHERE order\_no = [ORIGINAL ORDER] AND line\_item\_no = [ORIGINAL LINE ITEM] AND line\_item\_coil\_no =[ORIGINAL LINE ITEM COIL NO]

Where:

[ORIGINAL ORDER] is the order where the slab was originally assigned (before the reapplication)

[ORIGINAL LINE ITEM] is the original line item where the slab was originally assigned (before the reapplication)

[ORIGINAL LINE COIL NO] is the original line coil number where the slab was originally assigned (before the reapplication)

* + - * + Validate if the coil status from the original coil is in “RD” (Requires Disposition), if it does:

Begin an SQL Transaction

Update the slab list to original:

UPDATE mps\_slab\_list SET hsm\_schedule\_no = orig\_hsm\_schedule\_no , sched\_slab\_seq = orig\_sched\_slab\_seq,

sched\_heat\_seq =[ORIGINAL HEAT], sched\_string\_seq = [ORIGINAL SCHEDULE] ,

,order\_no = [ORIGINAL ORDER]

, line\_item\_no = [ORIGINAL LINE ITEM NUMBER]

, line\_item\_coil\_no = [LINE ITEM COIL NUMBER]

,stock\_id = [ORIGINAL STOCK]

, mill\_width\_aim = [ORIGINAL WIDTH AIM]

, mill\_gauge\_aim = [ORIGINAL GAUGE AIM]

, mill\_weight\_aim = [ORIGINAL WEIGHT AIM]

, slab\_thickness = [ORIGINAL SLAB THICKNESS]

, slab\_width\_head = [ORIGINAL SLAB WIDTH HEAD]

, slab\_width\_tail = [CURRENT SLAB WIDTH TAIL]

, slab\_length = [ORIGINAL SLAB LENGTH]

,coil\_length= [ORIGINAL COIL LENGTH]

,casting\_time=[ORIGINAL CASTING TIME]

,casting\_speed=[ORIGINAL CASTING SPEED]

,planned\_slab\_type=[ORIGINAL PLANNED SLAB TYPE]

Slab\_width\_change\_mode=[ORIGINAL WIDTH CHANGE MODE]

, mill\_weight\_min = [ORIGINAL COIL MIN WEIGHT]

, mill\_weight\_max = [ORIGINAL COIL MAX WEIGHT]

, coil\_length =[ORIGINAL COIL LENGTH]

WHERE mill\_prod\_slab\_seq = [ORIGINAL SLAB SEQUENCE]

* + - * + Where:

[ORIGINAL HEAT] is the heat that the slab had before it was reapplied,

[ORIGINAL SCHEDULE] is the schedule in which the slab was assigned before it was reapplied

[ORIGINAL ORDER] is the order in which the slab was assigned before it was reapplied

[ORIGINAL LINE ITEM NUMBER] is the current line item in which the slab was assigned before it was reapplied

[ORGINAL ITEM COIL NUMBER] is the coil number of the coil in which the slab was assigned before it was reapplied

[ORGINAL STOCK] is the original stock in which the slab was assigned before it was reapplied

[ORIGINAL WIDTH AIM] is the aim width in which the slab was assigned before it was reapplied

[ORIGINAL GAUGE AIM] is the gauge aim in which the slab was assigned before it was reapplied

[ORIGINAL WEIGHT AIM] is the weight aim in which the slab was assigned before it was reapplied

[ORIGINAL SLAB THICKNESS] is the current thickness in which the slab was assigned before it was reapplied

[ORIGINAL SLAB WIDTH HEAD] is the current slab width on the head in which the slab was assigned before it was reapplied

[ORIGINAL SLAB WIDTH TAIL] is the current slab wdth in which the slab was assigned before it was reapplied

[ORIGINAL SLAB LENGTH] is the slab length in which the slab was assigned before it was reapplied

[ORIGINAL COIL LENGTH] is the coil length in which the slab was assigned before it was reapplied

[CASTING TIME] is the casting time in which the slab was assigned before it was reapplied

[CASTING SPEED] is the casting speed in which the slab was assigned before it was reapplied

[ORIGINAL SLAB WIDTH CHANGE MODE] is the configuration the slab had on the width change before it was reapplied

[ORIGINAL COIL MIN WEIGHT] is the minimum weight in which the slab was assigned before it was reapplied

[ORIGINAL COIL MAX WEIGHT] is the maximum weight in which the slab was assigned before it was reapplied

[ORIGINAL COIL LENGTH] is the coil length in which the slab was assigned before it was reapplied

[ORIGINAL SLAB SEQUENCE] is the slab sequence number in which the slab was assigned before it was reapplied

* + - * Update the production comment :
        + UPDATE caster\_string\_slab SET production\_comments = '[COMMENTS]' WHERE hsm\_schedule\_no = [ORIGINAL SCHEDULE] AND sched\_slab\_seq =[ORIGINAL SLAB SCHEDULE SEQUENCE]
        + Where:

[ORIGINAL SCHEDULE] is the schedule in which the slab was assigned before it was reapplied

[ORIGINAL SLAB SCHEDULE SEQUENCE] is the sequence for the slab in which the slab was assigned before it was reapplied

* + - * Update the Coil Status:
        + UPDATE coil SET coil\_status = 'IS' WHERE order\_no =[ORIGINAL ORDER] AND line\_item\_no = [ORIGINAL LINE ITEM] AND line\_item\_coil\_no = [ORIGINAL LINE ITEM]
        + If the changeDateTime is not empty, then add this statement to the query: “AND change\_datetime=[CHANGE DATE TIME]
        + Where:

[ORIGINAL LINE ITEM] is the line item of the in which the slab was assigned before it was reapplied

[ORIGINAL ORDER] is the order of the in which the slab was assigned before it was reapplied

* + - * + [CHANGE DATE TIME] is the date and time in which the update is performed
      * Update the Line Item Status:

UPDATE customer\_order\_line\_item SET order\_no = order\_no WHERE order\_no = [CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM]

Where:

[CURRENT ORDER] is the order for the slab to be reapplied

[CURRENT LINE ITEM] is the line item number of the slab to be reapplied

* + - * Validate if the slab is stock and is planned. If it is stock and is planned:
        + Update the Coil Status:

UPDATE coil SET coil\_status = 'RD' WHERE order\_no =[ORIGINAL ORDER] AND line\_item\_no = [ORIGINAL LINE ITEM] AND line\_item\_coil\_no = [ORIGINAL LINE ITEM]

If the changeDateTime is not empty, then add this statement to the query: “AND change\_datetime=[CHANGE DATE TIME]

* + - * + Where:

[ORIGINAL LINE ITEM] is the line item of the in which the slab was assigned before it was reapplied

[ORIGINAL ORDER] is the order of the in which the slab was assigned before it was reapplied

* + - * + [CHANGE DATE TIME] is the date and time in which the update is performed
      * Update the Line Item Status:

UPDATE customer\_order\_line\_item SET order\_no = order\_no WHERE order\_no = [CURRENT ORDER] AND line\_item\_no = [CURRENT LINE ITEM]

Where:

[CURRENT ORDER] is the order for the slab to be reapplied

* + - * + [CURRENT LINE ITEM] is the line item number of the slab to be reapplied
  + If the slab is not on the list, refresh the list, and start refreshing the screen.

# UC072 - Slab Re-Application - Resend Schedule

## Description

This functionality allows the user to re-send the schedule to the Rolling Mills Level 2 system.

## Triggers

* When the user clicks on the “Re-send Schedule” button.

## Preconditions

* The user is on the Slab Re-Application Dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the slab that he wants to revert.

## The actor clicks on the “Re-send Schedule” button

## The system sends the schedule to the Rolling Mill L2 System (BR1).

## End of use case

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## BR1 – Send Message to L2

## Message from UC016 is sent.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* Send a message DEC\_L3\_ALL\_COILS (SR001VW.CPP Line 930)

# UC073 – Coil Rework - Refresh Yard

## Description

This functionality will allow the actor to refresh the coil number that is currently sitting in the coil inspection yard location.

## Triggers

* When the user clicks on the “Refresh Yard” button.

## Preconditions

* The user is on the Coil Re-Work dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Refresh Yard” button

## The system retrieves the coil on the inspection yard and is displayed on the screen (BR1).

## End of use case

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## BR1 – Display coil on inspection yard

## There is only one inspection yard, and it can hold just one coil at a time.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Production Coil Number | Output | Yes | Empty | N/A | coil\_yard\_locations | Production\_coil\_no |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to execute step 43.5.1.2 on the main flow, the following queries must be executed (Form.h Line 379):
  + SELECT production\_coil\_no FROM coil\_yard\_locations WHERE yard = [CURRENT YARD] AND column = [CURRENT COLUMN] AND row = [CURRENT ROW]
  + Where:
    - SELECT yard,column, row FROM special\_yard\_locations WHERE id = 'IN'
    - [CURRENT YARD]= yard field from previous query
    - [CURRENT COLUMN] column field from previous query
    - [CURRENT ROW] row field from previous query

# UC074 - Coil Rework – Input Mandatory Checks

## Description

This functionality allows the user to set the parameters to be validated when a coil is on the inspection yard.

## Triggers

* When the user modifies any of the fields on the “Mandatory Checks” section.

## Preconditions

* The user is on the Coil Re-Work dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor changes any selection or data from the “Mandatory Checks” section on the Coil Re-work dashboard

## The system updates the parameters for the mandatory checks (BR1).

## End of use case

## Alternative and Exception Flows

## There are no alternative or exception flows

## Business Rules

## BR1 – Display coil on inspection yard

## The mandatory checks must validate the following:

## One Radio button field with 3 options:

## Yes: the coil width is validated against the database

## No: the coil width and gauge are validated against the database

## Test Weight: no values are checked

## Input Width: actual coil width

## Input Gauge actual coil gauge.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Is coil number Verified | Input | Yes | Empty | Radio Button with three options: YES, NO, Test Weight |  |  |
| Input width | Input | Yes | Empty | Numeric, one decimal place. |  |  |
| Input Gauge | Input | Yes | Empty | Numeric, one decimal place. |  |  |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

This use case explains the functionality of the radio buttons, but does not affect the system until the send button is clicked.

# UC075 - Coil Rework – Send

## Description

This functionality allows the user to send the coil back to Level 2, while in rework location.

## Triggers

* When the user clicks on the “Send” button.

## Preconditions

* The user is on the Coil Re-Work dashboard

## Actors

* User

## Basic Flow

## Basic Flow

## The actor inputs the width and gauge data on the dashboard

## The actor clicks on the “Send” button

## The system validates the information entered (BR1) (AF1).

## The sends the coil information to Level 2 system (BR2).

## End of use case

## Alternative and Exception Flows

## AF1 – Validation Failed

## The selected validation failed, and the system shows an error message

## End of use case.

## Business Rules

## BR1 – Display coil on inspection yard

## The mandatory checks must validate the following:

## Yes: the coil width is validated against the database

## No: the coil width and gauge are validated against the database

## Test Weight: no values are checked

## Input Width: actual coil width

## Input Gauge actual coil gauge.

## BR2 – Send Message to L2

## Message from UC030 is sent to L2.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
| Is coil number Verified | Input | Yes | Empty | Radio Button with three options: YES, NO, Test Weight |
| Input width | Input | Yes | Empty | Numeric, one decimal place. |
| Input Gauge | Input | Yes | Empty | Numeric, one decimal place. |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

* In order to perform step 45.5.1.3 (Form1.cs Line 68) the following validations must be performed:
  + Disable the “Send” button
  + Validate that the coil on the yard is different than 0, if it is, continue with the next step. If its equal than 0, that means that there is no coil on the yard, and an error must be generated.
    - SELECT production\_coil\_no FROM coil\_yard\_locations WHERE yard = [CURRENT YARD] AND column = [CURRENT COLUMN] AND row = [CURRENT ROW]
  + If the production coil number is equal to 0, then the position is deleted, an finish the execution.
  + Validate which of the “is coil number verified” radio buttons is selected:
    - If “No” is selected:
      * Retrieve the width and gauge data from the text fields.
      * Retrieve the coil measurements for gauge and width:
        + SELECT actual\_width\_avg, actual\_thickness\_avg FROM coil\_product\_data WHERE production\_coil\_no = [CURRENT COIL]
      * Evaluate the actual coil measurements on the database against the data entered by the user:
        + Add 0.1 to the width and gauge obtained from the database, to set the maximum width and gauge.
        + Subtract 0.1 to the width and gauge obtained from the database, to set the minimum width and gauge.
        + Compare the width entered by the user against the calculated maximum width and minimum width:

If the width entered by the user is greater than or equal than the minimum width, and is less than or equal than the maximum width, then the width test is successful.

If the gauge entered by the user is greater than or equal than the minimum gauge, and is less than or equal than the maximum gauge, then the gauge test is successful.

* + - * + If the width and gauge test failed, send an error message must be displayed to the user.
        + Enable the send button.
    - If “Yes” was selected:
      * Retrieve the width data from the text fields.
      * Retrieve the coil measurements for gauge and width:
        + SELECT actual\_width\_avg, actual\_thickness\_avg FROM coil\_product\_data WHERE production\_coil\_no = [CURRENT COIL]
      * Evaluate the actual coil measurements on the database against the data entered by the user:
        + Add 0.1 to the width obtained from the database, to set the maximum width.
        + Subtract 0.1 to the width obtained from the database, to set the minimum width.
        + Compare the width entered by the user against the calculated maximum width and minimum width:

If the width entered by the user is greater than or equal than the minimum width, and is less than or equal than the maximum width, then the width test is successful.

* + - * + If the width test failed, send an error message must be displayed to the user.
      * Enable the send button.
    - If the “Test Weight” is selected:
      * Nothing is compared.
  + If the result of the tests is true, a message from UC030 is sent to L2.
  + Delete the coil position:
    - Begin SQL Transaction
    - DELETE FROM coil\_yard\_locations sbSelect.Append("WHERE production\_coil\_no = [COIL IN YARD]
    - Update the coil status:
      * UPDATE coil\_product\_data SET coil\_yard\_status = 'RE',change\_datetime = getdate(), change\_user\_id = [CURRENT USER] WHERE production\_coil\_no = [COIL IN YARD]
      * UPDATE coil\_yard\_rows SET vacancy = vacancy + 1, change\_datetime = getdate(),change\_user\_id = [CURRENT USER] WHERE yard = [CURRENT YARD] AND column = [CURRENT COLUMN] AND row =[CURRENT ROW]
    - Commit the transaction

# UC076 - Delays – Record Delay (manually)

## Description

This functionality allows the user enter a delay on an screen.

## Triggers

* When the user clicks on the “Delays” button.

## Preconditions

* The user is on the Mill Schedule Dashboard or in the Slab Re-Apply Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor clicks on the “Delays” button

## The system opens a modal window with the following data to be entered:

## Area

## Discipline

## Equipment Id

## Equipment Name

## Planned

## Start Time

## End Time

## Duration

## Crew

## Sequence Break

## Cobble

## Reason

## EAF Stop Time

## Sequence

## Starting Heat

## Ending Heat

## Aim Vc

## Slowest Vc Slow

## Fastest Vc Slow

## Width

## The actor clicks enters the required information (BR1)

## The system validates the information entered (BR2) (AF1)

## The system registers the delay.

## End of use case

## Alternative and Exception Flows

## AF1 – Validation Failed

## The selected validation failed, and the system shows an error message

## End of use case.

## Business Rules

## BR1 – Data Entry Validations:

## Start date must be before end date

## Duration cannot be negative

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Area | Input | Yes | Empty | Combo Box. Must show the available areas. | Delays | AREA |
| Discipline | Input | Yes | Empty | Combo Box. Must show the available disciplines. | Delays | DISCIPLINE |
| Equipment Id | Input | Yes | Empty | Combo box. Must show the available equipment. | Delays | EQUIPMENT\_ID |
| Equipment Name | Input | Yes | Empty | Must display the description from the selected equipment id | Delays | EQUIPMENT\_NAME |
| Planned | Input | Yes | Empty | N/A | Delays | PLANNED |
| Start Time | Input | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS | Delays | START\_TIME |
| End Time | Input | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS | Delays | END\_TIME |
| Duration | Input | Yes | Empty | Numeric, integer | Delays | DURATION |
| Crew | Input | Yes | Empty | Combo box with the available crews. | Delays | CREW |
| Sequence Break | Input | Yes | Empty | Checkbox | Delays | SEQUENCE\_BREAK |
| Cobble | Input | Yes | Empty | Checkbox | Delays | COBBLE |
| Reason | Input | Yes | Empty | N/A | Delays | REASON |
| EAF Stop Time | Input | Yes | Empty | N/A | Delays | EAF\_STOP\_TIME |
| Sequence | Input | Yes | Empty | Numeric integer | Delays | SEQUENCE |
| Starting Heat | Input | Yes | Empty | Numeric integer | Delays | STARTING\_HEAT |
| Ending Heat | Input | Yes | Empty | Numeric integer | Delays | ENDING\_HEAT |
| Aim Vc | Input | Yes | Empty | Numeric, one decimal | Delays | AIM\_VC |
| Slowest Vc Slow | Input | Yes | Empty | Numeric, one decimal | Delays | SLOWEST\_VC\_SLOW |
| Fastest Vc Slow | Input | Yes | Empty | Numeric, one decimal | Delays | FASTEST\_VV\_SLOW |
| Width | Input | Yes | Empty | Numeric integer | Delays | WIDTH |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* This information must be stored at the SQL SERVER database.

# UC077 - Delays – Record Delay (automatic)

## Description

This functionality allows to the user to record a delay automatically, and then specify the delay reason and additional data.

## Triggers

* When a delay occurs in the plant and is registered on the Level 2 system.

## Preconditions

* The user is on the Mill Schedule Dashboard or in the Slab Re-Apply Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## A delay is registered on the Level 2 system.

## The system shows a message that notifies the user that there is a delay registered on the Level 2 system

## The actor clicks on the “Delays” icon

## The system shows the delays window, with the following delay prefilled:

## Start Time

## End Time

## Duration

## The actor enters the remaining data which is the following:

## Area

## Discipline

## Equipment Id

## Equipment Name

## Planned

## Crew

## Sequence Break

## Cobble

## Reason

## EAF Stop Time

## Sequence

## Starting Heat

## Ending Heat

## Aim Vc

## Slowest Vc Slow

## Fastest Vc Slow

## Wdith

## The actor enters the required information (BR1)

## The system validates the information entered (BR2) (AF1)

## The system registers the delay.

## End of use case

## Alternative and Exception Flows

## AF1 – Validation Failed

## The selected validation failed, and the system shows an error message

## End of use case.

## Business Rules

## BR1 – Data Entry Validations:

## Start date must be before end date

## Duration cannot be negative

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Area | Input | Yes | Empty | Combo Box. Must show the available areas. | Delays | AREA |
| Discipline | Input | Yes | Empty | Combo Box. Must show the available disciplines. | Delays | DISCIPLINE |
| Equipment Id | Input | Yes | Empty | Combo box. Must show the available equipment. | Delays | EQUIPMENT\_ID |
| Equipment Name | Input | Yes | Empty | Must display the description from the selected equipment id | Delays | EQUIPMENT\_NAME |
| Planned | Input | Yes | Empty | N/A | Delays | PLANNED |
| Start Time | Input | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the delay start time from L2 | Delays | START\_TIME |
| End Time | Input | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the delay end time from L2 | Delays | END\_TIME |
| Duration | Input | Yes | Empty | Numeric, integer. Prefilled with the delay duration from L2 | Delays | DURATION |
| Crew | Input | Yes | Empty | Combo box with the available crews. | Delays | CREW |
| Sequence Break | Input | Yes | Empty | Checkbox | Delays | SEQUENCE\_BREAK |
| Cobble | Input | Yes | Empty | Checkbox | Delays | COBBLE |
| Reason | Input | Yes | Empty | N/A | Delays | REASON |
| EAF Stop Time | Input | Yes | Empty | N/A | Delays | EAF\_STOP\_TIME |
| Sequence | Input | Yes | Empty | Numeric integer | Delays | SEQUENCE |
| Starting Heat | Input | Yes | Empty | Numeric integer | Delays | STARTING\_HEAT |
| Ending Heat | Input | Yes | Empty | Numeric integer | Delays | ENDING\_HEAT |
| Aim Vc | Input | Yes | Empty | Numeric, one decimal | Delays | AIM\_VC |
| Slowest Vc Slow | Input | Yes | Empty | Numeric, one decimal | Delays | SLOWEST\_VC\_SLOW |
| Fastest Vc Slow | Input | Yes | Empty | Numeric, one decimal | Delays | FASTEST\_VV\_SLOW |
| Width | Input | Yes | Empty | Numeric integer | Delays | WIDTH |
| Event Type | Input | Yes | 1 | Numeric Integer | Delays | EVENT\_TYPE |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to obtain the delays from the INSQL database, the following query must be performed:
  + Obtain the delay events from the insql, for a specific date range. Query the following tags:
    - AW7216\_SI-001
    - AW7416
    - AW7404
    - MLC\_EVENT\_MODE\_AUTO
    - A117784
    - A117763
    - A117761
    - SS\_AutoStop
    - SS\_AutoStart
    - L2\_HT\_ID\_OUT
    - AW6694
    - AW7196
    - A11805F
    - A64430
    - MLC\_EVENT\_MODE\_AUTOSTART
  + From the table “History” with retrieval mode cyclic
  + Validate from the results of the query, if:
    - (MLC\_EVENT\_MODE\_AUTO>0 OR SS\_Autostart >0 OR MLC\_EVENT\_MODE\_AUTOSTART>0) AND StopperPosition>3.0 AND CastMode>0 and Simulation<1
  + If its true, then:
    - Post a delay (postdelay++)
    - Validate if SSAutostop>0 or tailoutmode>0. If it is:
      * Status =”Tailing Out”
    - If not, validate if postdelay<28, if true
      * Status= “ Startup Rampup”
    - Else
      * Status=casting
      * Validate if the l2heatnumber>0. If it is:
        + Validate that there is a grade specified. If it is:

Look into the speed table to find the width, for each speed row:

Slowcount++

Speed on the VcPv from the grade table multiplied by 2

Min speed is the maxVC

Validate if the speed on the table plus .1 is less than the min Speed. Validate if:

The slowCount>1

Status=”Slowdown”

If not

Status=”< 1 minute”

If the speed is more than the min speed:

Slowcount=0;

# UC078 - Delays – Assign a Delay Reason

## Description

This functionality allows to the user to assign a delay reason to any delay event that has been registered.

## Triggers

* When the actor clicks on the “Assign Delay Reason” button.

## Preconditions

* The user is on the Mill Schedule Dashboard or in the Slab Re-Apply Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The user clicks on the “Assign Delay Reason” button.

## The system displays a list of the existing delays for the last 12 hours, with the following data:

* + Area
  + Discipline
  + Equipment Id
  + Equipment Description
  + Start Time
  + End Time
  + Duration
  + Crew
  + Reason

## The actor selects one of the delays on the list and clicks on the “Assign Delay Reason” button

## The system displays a modal window with the following fields to be entered by the user:

## Discipline

## Reason

## The actor clicks in the “Assign” button (AF1)

## The system stores the delay reason

## End of use case.

## Alternative and Exception Flows

## AF1 – Clicks on the cancel button

## The system closes the modal window, no data is saved.

## End of use case.

## Business Rules

## No business rules required for this use case

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Area | Output | Yes | Empty | Combo Box. Must show the available areas. | Delays | AREA |
| Equipment Id | Output | Yes | Empty | Combo box. Must show the available equipment. | Delays | EQUIPMENT\_ID |
| Equipment Name | Output | Yes | Empty | Must display the description from the selected equipment id | Delays | EQUIPMENT\_NAME |
| Start Time | Output | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the delay start time from L2 | Delays | START\_TIME |
| End Time | Output | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the delay end time from L2 | Delays | END\_TIME |
| Duration | Output | Yes | Empty | Numeric, integer. Prefilled with the delay duration from L2 | Delays | DURATION |
| Crew | Output | Yes | Empty | Combo box with the available crews. | Delays | CREW |
| Reason | Input | Yes | Empty | N/A | Delays | REASON |
| Discipline | Input | Yes | Empty | Combo Box. Must show the available disciplines. | Delays | DISCIPLINE |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specifications

* Discipline and Reason combo boxes will be loaded with information from the master data catalogs.
* Data will be stored on the SQL Server database.

# UC079 - Delays – Split a Delay Event

## Description

This functionality allows to the user split a delay event into two so he can assign different reasons to each part of the total duration.

## Triggers

* When the actor clicks on the “Split” button.

## Preconditions

* The user is on the Mill Schedule Dashboard or in the Slab Re-Apply Dashboard, entered into the “Assign Reason” option.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects one of the delays on the list and clicks on the “Split Delay” button

## The system displays a modal window with the following fields (BR1):

## Start Time

## Split Time

## End Time

## The actor enters the split time (BR2) (AF1) and clicks on the “Split” button.

## The system updates the delay and creates a new delay (BR3).

## End of use case.

## Alternative and Exception Flows

## AF1 – Clicks on the cancel button

## The system closes the modal window, no data is saved.

## End of use case.

## Business Rules

## BR1 – Prefilled Fields on the Split Delay

## The Start Time and End Time must be prefilled with the start time and end time.

## BR2 – Split Time Validation

## The time selected to split must be after and not equal the start time and before and not equal the end time.

## BR3 – Store Delays on Split

## The system must update the end time of the selected delay with the split date, and insert a new delay with the split date as the start date, and the end date of the selected delay as the end date.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Start Time | Output | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the delay start time from L2 | Delays | START\_TIME |
| End Time | Output | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the delay end time from L2 | Delays | END\_TIME |
| Split Time | INput | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Apply BR2 |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

None

# UC080 - Delays – Record Slowdown (automatic)

## Description

This functionality allows to the user to record a slowdown automatically, and then specify the slowdown reason and additional data.

## Triggers

* When a slowdown occurs in the plant and is registered on the Level 2 system.

## Preconditions

* The user is on the Mill Schedule Dashboard or in the Slab Re-Apply Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## A slowdown is registered on the Level 2 system (BR1).

## The system shows a message that notifies the user that there is a slowdown registered on the Level 2 system

## The actor clicks on the “Slowdowns” icon

## The system shows the slowdowns window, with the following delay prefilled:

## Start Time

## End Time

## Duration

## The actor enters the remaining data which is the following:

## Area

## Discipline

## Equipment Id

## Equipment Name

## Planned

## Crew

## Sequence Break

## Cobble

## Reason

## EAF Stop Time

## Sequence

## Starting Heat

## Ending Heat

## Aim Vc

## Slowest Vc Slow

## Fastest Vc Slow

## Width

## The actor enters the required information (BR2)

## The system validates the information entered (BR3) (AF1)

## The system registers the delay.

## End of use case

## Alternative and Exception Flows

## AF1 – Validation Failed

## The selected validation failed, and the system shows an error message

## End of use case.

## Business Rules

## BR1 – Determine if there is a Slowdown

## An estimated caster time will be calculated from the L2 database, and compared against the standard times. If the time is more than the standard time, a slowdown will be recorded.

## BR2 – Data Entry Validations:

## Start date must be before end date

## Duration cannot be negative

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Area | Input | Yes | Empty | Combo Box. Must show the available areas. | Delays | AREA |
| Discipline | Input | Yes | Empty | Combo Box. Must show the available disciplines. | Delays | DISCIPLINE |
| Equipment Id | Input | Yes | Empty | Combo box. Must show the available equipment. | Delays | EQUIPMENT\_ID |
| Equipment Name | Input | Yes | Empty | Must display the description from the selected equipment id | Delays | EQUIPMENT\_NAME |
| Planned | Input | Yes | Empty | N/A | Delays | PLANNED |
| Start Time | Input | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the slowdown start time from L2 | Delays | START\_TIME |
| End Time | Input | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the slowdown end time from L2 | Delays | END\_TIME |
| Duration | Input | Yes | Empty | Numeric, integer. Prefilled with the slowdown duration from L2 | Delays | DURATION |
| Crew | Input | Yes | Empty | Combo box with the available crews. | Delays | CREW |
| Sequence Break | Input | Yes | Empty | Checkbox | Delays | SEQUENCE\_BREAK |
| Cobble | Input | Yes | Empty | Checkbox | Delays | COBBLE |
| Reason | Input | Yes | Empty | N/A | Delays | REASON |
| EAF Stop Time | Input | Yes | Empty | N/A | Delays | EAF\_STOP\_TIME |
| Sequence | Input | Yes | Empty | Numeric integer | Delays | SEQUENCE |
| Starting Heat | Input | Yes | Empty | Numeric integer | Delays | STARTING\_HEAT |
| Ending Heat | Input | Yes | Empty | Numeric integer | Delays | ENDING\_HEAT |
| Aim Vc | Input | Yes | Empty | Numeric, one decimal | Delays | AIM\_VC |
| Slowest Vc Slow | Input | Yes | Empty | Numeric, one decimal | Delays | SLOWEST\_VC\_SLOW |
| Fastest Vc Slow | Input | Yes | Empty | Numeric, one decimal | Delays | FASTEST\_VV\_SLOW |
| Width | Input | Yes | Empty | Numeric integer | Delays | WIDTH |
| Event Type | Input | Yes | 2 | Numeric Integer | Delays | EVENT\_TYPE |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

* In order to obtain the delays from the INSQL database, the following query must be performed:
  + Obtain the delay events from the insql, for a specific date range. Query the following tags:
    - AW7216\_SI-001
    - AW7416
    - AW7404
    - MLC\_EVENT\_MODE\_AUTO
    - A117784
    - A117763
    - A117761
    - SS\_AutoStop
    - SS\_AutoStart
    - L2\_HT\_ID\_OUT
    - AW6694
    - AW7196
    - A11805F
    - A64430
    - MLC\_EVENT\_MODE\_AUTOSTART
  + From the table “History” with retrieval mode cyclic
  + Validate from the results of the query, if:
    - (MLC\_EVENT\_MODE\_AUTO>0 OR SS\_Autostart >0 OR MLC\_EVENT\_MODE\_AUTOSTART>0) AND StopperPosition>3.0 AND CastMode>0 and Simulation<1
  + If its true, then:
    - Post a delay (postdelay++)
    - Validate if SSAutostop>0 or tailoutmode>0. If it is:
      * Status =”Tailing Out”
    - If not, validate if postdelay<28, if true
      * Status= “ Startup Rampup”
    - Else
      * Status=casting
      * Validate if the l2heatnumber>0. If it is:
        + Validate that there is a grade specified. If it is:

Look into the speed table to find the width, for each speed row:

Slowcount++

Speed on the VcPv from the grade table multiplied by 2

Min speed is the maxVC

Validate if the speed on the table plus .1 is less than the min Speed. Validate if:

The slowCount>1

Status=”Slowdown”

If not

Status=”< 1 minute”

If the speed is more than the min speed:

* Slowcount=0

# UC081 - Delays – Assign a Slowdown Reason

## Description

This functionality allows to the user to assign a slowdown reason to any slowdown event that has been registered.

## Triggers

* When the actor clicks on the “Assign Slowdown Reason” button.

## Preconditions

* The user is on the Mill Schedule Dashboard or in the Slab Re-Apply Dashboard.

## Actors

* User

## Basic Flow

## Basic Flow

## The user clicks on the “Assign Slowdown Reason” button.

## The system displays a list of the existing slowdowns for the las 12 hours, with the following data:

* + Area
  + Discipline
  + Equipment Id
  + Equipment Description
  + Start Time
  + End Time
  + Duration
  + Crew
  + Reason

## The actor selects one of the slowdowns on the list and clicks on the “Assign Slowdown Reason” button

## The system displays a modal window with the following fields to be entered by the user:

## Discipline

## Reason

## The actor clicks in the “Assign” button (AF1)

## The system stores the slowdown reason

## End of use case.

## Alternative and Exception Flows

## AF1 – Clicks on the cancel button

## The system closes the modal window, no data is saved.

## End of use case.

## Business Rules

## No business rules required for this use case

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Area | Output | Yes | Empty | Combo Box. Must show the available areas. | Delays | AREA |
| Equipment Id | Output | Yes | Empty | Combo box. Must show the available equipment. | Delays | EQUIPMENT\_ID |
| Equipment Name | Output | Yes | Empty | Must display the description from the selected equipment id | Delays | EQUIPMENT\_NAME |
| Start Time | Output | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the slowdown start time from L2 | Delays | START\_TIME |
| End Time | Output | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the slowdown end time from L2 | Delays | END\_TIME |
| Duration | Output | Yes | Empty | Numeric, integer. Prefilled with the slowdown duration from L2 | Delays | DURATION |
| Crew | Output | Yes | Empty | Combo box with the available crews. | Delays | CREW |
| Reason | Input | Yes | Empty | N/A | Delays | REASON |
| Discipline | Input | Yes | Empty | Combo Box. Must show the available disciplines. | Delays | DISCIPLINE |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

Slowdown reasons and disciplines combo box will be populated with information from the catalog.

# UC082 - Delays – Split a Slowdown Event

## Description

This functionality allows to the user split a Slowdown event into two so he can assign different reasons to each part of the total duration.

## Triggers

* When the actor clicks on the “Split” button.

## Preconditions

* The user is on the Mill Schedule Dashboard or in the Slab Re-Apply Dashboard, entered into the “Assign Reason” option.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects one of the Slowdowns on the list and clicks on the “Split Slowdown” button

## The system displays a modal window with the following fields (BR1):

## Start Time

## Split Time

## End Time

## The actor enters the split time (BR2) (AF1) and clicks on the “Split” button.

## The system updates the Slowdown and creates a new Slowdown (BR3).

## End of use case.

## Alternative and Exception Flows

## AF1 – Clicks on the cancel button

## The system closes the modal window, no data is saved.

## End of use case.

## Business Rules

## BR1 – Prefilled Fields on the Split Slowdown

## The Start Time and End Time must be prefilled with the start time and end time.

## BR2 – Split Time Validation

## The time selected to split must be after and not equal the start time and before and not equal the end time.

## BR3 – Store Slowdowns on Split

## The system must update the end time of the selected Slowdown with the split date, and insert a new Slowdown with the split date as the start date, and the end date of the selected Slowdown as the end date.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Start Time | Output | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the Slowdown start time from L2 | Delays | START\_TIME |
| End Time | Output | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Prefilled with the Slowdown end time from L2 | Delays | END\_TIME |
| Split Time | INput | Yes | Empty | Format MM/dd/YYYY HH24:mm:SS. Apply BR2 |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Notes

None

# UC083 – Master Data – Delay Reasons Maintenance

## Description

This functionality allows to the user to add, update or delete a delay reason.

## Triggers

* When the actor clicks on the “Delays Reasons Maintenance” button.

## Preconditions

* None.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the “Delays Reasons Maintenance Option”

## The system displays a list with the following information:

## Delay Reason Name

## Delay Reason Description

## The actor enters the following information to create a new delay reason:

## Delay Reason Name

## Delay Reason Description

## The actor clicks on the “Save” button (AF1)(AF2)(AF3)

## The system stores the new delay reason.

## End of use case.

## Alternative and Exception Flows

## AF1 – Clicks on the Update button

## The actor select one of the existing delay reason from the list

## The system display the information of the selected delay reason on the data fields

## The actor enters the new data for the delay reason and clicks on the “Update” button

## The system updates the delay reason data.

## End of use case.

## AF2 – Clicks on the Delete button

## The actor select one of the existing delay reason from the list

## The system display the information of the selected delay reason on the data fields

## The actor clicks on the “Delete” button (BR1)

## The system display a confirmation message to delete the delay reason

## The user click on the “OK” button (AF3)

## The system deletes delay reason.

## End of use case.

## AF3 – Clicks on the cancel button

## The system cancel the action

## End of use case.

## Business Rules

## BR1 – Delete Validation

## The system must validate that the delay reason is not used in any delay event, if it’s used, it cannot be deleted.

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Delay Reason Name | In/Out | Yes | Empty | String, no symbols. | DELAY\_RSN | DELAY\_RSN\_NAME |
| Delay Reason Description | In/out | Yes | Empty | N/A | DELAY\_RSN | DELAY\_RSN\_DESC |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

Data must be stored on the SQL SERVER database.

# UC084 – Master Data – Slowdown Reasons Maintenance

## Description

This functionality allows to the user to add, update or delete a Slowdown reason.

## Triggers

* When the actor clicks on the “Slowdowns Reasons Maintenance” button.

## Preconditions

* None.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the “Slowdowns Reasons Maintenance Option”

## The system displays a list with the following information:

## Slowdown Reason Name

## Slowdown Reason Description

## The actor enters the following information to create a new Slowdown reason:

## Slowdown Reason Description

## The actor clicks on the “Save” button (AF1)(AF2)(AF3)

## The system stores the new Slowdown reason.

## End of use case.

## Alternative and Exception Flows

## AF1 – Clicks on the Update button

## The actor select one of the existing Slowdown reason from the list

## The system display the information of the selected Slowdown reason on the data fields

## The actor enters the new data for the Slowdown reason and clicks on the “Update” button

## The system updates the Slowdown reason data.

## End of use case.

## AF2 – Clicks on the Delete button

## The actor select one of the existing Slowdown reason from the list

## The system display the information of the selected Slowdown reason on the data fields

## The actor clicks on the “Delete” button (BR1)

## The system display a confirmation message to delete the Slowdown reason

## The user click on the “OK” button (AF3)

## The system deletes Slowdown reason.

## End of use case.

## AF3 – Clicks on the cancel button

## The system cancel the action

## End of use case.

## Business Rules

## BR1 – Delete Validation

## The system must validate that the Slowdown reason is not used in any Slowdown event, if it’s used, it cannot be deleted.

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Slowdown Reason Name | In/Out | Yes | Empty | String, no symbols. | SLOWDOWN\_RSN | SLOWDOWN\_RSN\_NAME |
| Slowdown Reason Description | In/out | Yes | Empty | N/A | SLOWDOWN\_RSN | SLOWDOWN\_RSN\_DESC |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

Data must be stored at the SQL SERVER database.

# UC085 – Master Data – Areas Maintenance

## Description

This functionality allows to the user to add, update or delete an Area.

## Triggers

* When the actor clicks on the “Area Maintenance” button.

## Preconditions

* None.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the “Area Maintenance Option”

## The system displays a list with the following information:

## Area Name

## Area Description

## The actor enters the following information to create a new Area:

## Area Description

## The actor clicks on the “Save” button (AF1)(AF2)(AF3)

## The system stores the new Area reason.

## End of use case.

## Alternative and Exception Flows

## AF1 – Clicks on the Update button

## The actor select one of the existing Area from the list

## The system display the information of the selected Area on the data fields

## The actor enters the new data for the Area and clicks on the “Update” button

## The system updates the Area data.

## End of use case.

## AF2 – Clicks on the Delete button

## The actor select one of the existing Area from the list

## The system display the information of the selected Area on the data fields

## The actor clicks on the “Delete” button (BR1)

## The system display a confirmation message to delete the Area

## The user click on the “OK” button (AF3)

## The system deletes the Area.

## End of use case.

## AF3 – Clicks on the cancel button

## The system cancel the action

## End of use case.

## Business Rules

## BR1 – Delete Validation

## The system must validate that the Area is not used in any slowdown or delay event, if it’s used, it cannot be deleted.

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Area Name | In/Out | Yes | Empty | String, no symbols. | AREA | AREA\_NAME |
| Area Description | In/out | Yes | Empty | N/A | AREA | AREA\_DESC |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

Data must be stored at the SQL Server Database.

# UC086 – Master Data – Disciplines Maintenance

## Description

This functionality allows to the user to add, update or delete a Discipline.

## Triggers

* When the actor clicks on the “Discipline Maintenance” button.

## Preconditions

* None.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the “Discipline Maintenance Option”

## The system displays a list with the following information:

## Discipline Name

## Discipline Description

## The actor enters the following information to create a new Discipline:

## Discipline Description

## The actor clicks on the “Save” button (AF1)(AF2)(AF3)

## The system stores the new Discipline reason.

## End of use case.

## Alternative and Exception Flows

## AF1 – Clicks on the Update button

## The actor select one of the existing Discipline from the list

## The system display the information of the selected Discipline on the data fields

## The actor enters the new data for the Discipline and clicks on the “Update” button

## The system updates the Discipline data.

## End of use case.

## AF2 – Clicks on the Delete button

## The actor select one of the existing Discipline from the list

## The system display the information of the selected Discipline on the data fields

## The actor clicks on the “Delete” button (BR1)

## The system display a confirmation message to delete the Discipline

## The user click on the “OK” button (AF3)

## The system deletes the Discipline.

## End of use case.

## AF3 – Clicks on the cancel button

## The system cancel the action

## End of use case.

## Business Rules

## BR1 – Delete Validation

## The system must validate that the Discipline is not used in any slowdown or delay event, if it’s used, it cannot be deleted.

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database Field** |
| Discipline Name | In/Out | Yes | Empty | Must be numeric, its an automatic generated number by the system. | DISCIPLINE | DISC\_NAME |
| Discipline Description | In/out | Yes | Empty | N/A | DISCIPLINE | DISC\_DESC |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

Data must be stored at the SQL Server database.

# UC087 – Master Data – Equipment Maintenance

## Description

This functionality allows to the user to add, update or delete an Equipment.

## Triggers

* When the actor clicks on the “Equipment Maintenance” button.

## Preconditions

* None.

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the “Equipment Maintenance Option”

## The system displays a list with the following information:

## Equipment name

## Equipment Description

## The actor enters the following information to create a new Equipment:

## Equipment Description

## The actor clicks on the “Save” button (AF1)(AF2)(AF3)

## The system stores the new Equipment reason.

## End of use case.

## Alternative and Exception Flows

## AF1 – Clicks on the Update button

## The actor select one of the existing Equipment from the list

## The system display the information of the selected Equipment on the data fields

## The actor enters the new data for the Equipment and clicks on the “Update” button

## The system updates the Equipment data.

## End of use case.

## AF2 – Clicks on the Delete button

## The actor select one of the existing Equipment from the list

## The system display the information of the selected Equipment on the data fields

## The actor clicks on the “Delete” button (BR1)

## The system display a confirmation message to delete the Equipment

## The user click on the “OK” button (AF3)

## The system deletes the Equipment.

## End of use case.

## AF3 – Clicks on the cancel button

## The system cancel the action

## End of use case.

## Business Rules

## BR1 – Delete Validation

## The system must validate that the Equipment is not used in any slowdown or delay event, if it’s used, it cannot be deleted.

## 

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database field** |
| Equipment name | In/Out | Yes | Empty | String, no symbols. | EQUIPMENT | EQ\_NAME |
| Equipment Description | In/out | Yes | Empty | N/A | EQUIPMENT | EQ\_DESC |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |
|  |  |  |  |  |  |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

Data must be stored at the SQL Server database.

# UC088 – Delays – Calculate Crew

## Description

This functionality creates the schedule for the different crews. This will allow to prefill the crew field on the delay and slowdown events.

## Triggers

* It’s a scheduled job that will run every week on Sunday 11:00 pm.

## Preconditions

* None

## Actors

* User

## Basic Flow

## Basic Flow

## The system calculates de crew

## End of use case.

## Alternative and Exception Flows

## No alternate or exception flows

## Business Rules

## No business rules required for this use case

## Input / Output Fields Properties and Validations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** |
|  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

To calculate the crew:

char CBaseMeltShopService::GetCrew(COleDateTime startTime)

{

       LogMessage(DEBUG\_LOG, "Enter GetCrew: startTime = %s", startTime.Format(MS\_TIMEFORMAT));

       char crew = 'E';

       if (startTime.GetStatus() == COleDateTime::valid)

       {

              COleDateTime base;

              base.SetDateTime(2001, 12, 23, 7, 0, 0);

              int span = 112;

              COleDateTimeSpan calc;

              int mod;

              calc = (startTime - base);

              int days, nightflub;

              if (startTime.GetHour() >= 19 || startTime.GetHour() < 7 )

              {

                     nightflub = 1;

              }

              else

              {

                     nightflub = 0;

              }

              days = calc.GetDays() \* 2 + nightflub;

              mod =  abs(days) % span;

              if (mod > 55)

              {

                     int switcher;

                     switcher = mod % 2;

                     mod -= 56;

                     if (switcher == 0)

                     {

                           mod += 1;

                     }

                     if (switcher == 1)

                     {

                           mod -= 1;

                     }

              }

              switch (abs(mod))

              {

                           case 0 :crew = 'B'; break;

                           case 1 :crew = 'C'; break;

                           case 2 :crew = 'A'; break;

                           case 3 :crew = 'D'; break;

                           case 4 :crew = 'A'; break;

                           case 5 :crew = 'D'; break;

                           case 6 :crew = 'B'; break;

                           case 7 :crew = 'C'; break;

                           case 8 :crew = 'B'; break;

                           case 9 :crew = 'C'; break;

                           case 10 :crew = 'A'; break;

                           case 11 :crew = 'D'; break;

                           case 12 :crew = 'A'; break;

                           case 13 :crew = 'D'; break;

                           case 14 :crew = 'A'; break;

                           case 15 :crew = 'D'; break;

                           case 16 :crew = 'C'; break;

                           case 17 :crew = 'B'; break;

                           case 18 :crew = 'C'; break;

                           case 19 :crew = 'B'; break;

                           case 20 :crew = 'A'; break;

                           case 21 :crew = 'D'; break;

                           case 22 :crew = 'A'; break;

                           case 23 :crew = 'D'; break;

                           case 24 :crew = 'C'; break;

                           case 25 :crew = 'B'; break;

                           case 26 :crew = 'C'; break;

                           case 27 :crew = 'B'; break;

                           case 28 :crew = 'C'; break;

                           case 29 :crew = 'B'; break;

                           case 30 :crew = 'A'; break;

                           case 31 :crew = 'D'; break;

                           case 32 :crew = 'A'; break;

                           case 33 :crew = 'D'; break;

                           case 34 :crew = 'C'; break;

                           case 35 :crew = 'B'; break;

                           case 36 :crew = 'C'; break;

                           case 37 :crew = 'B'; break;

                           case 38 :crew = 'A'; break;

                           case 39 :crew = 'D'; break;

                           case 40 :crew = 'A'; break;

                           case 41 :crew = 'D'; break;

                           case 42 :crew = 'A'; break;

                           case 43 :crew = 'D'; break;

                           case 44 :crew = 'C'; break;

                           case 45 :crew = 'B'; break;

                           case 46 :crew = 'C'; break;

                           case 47 :crew = 'B'; break;

                           case 48 :crew = 'A'; break;

                           case 49 :crew = 'D'; break;

                           case 50 :crew = 'A'; break;

                           case 51 :crew = 'D'; break;

                           case 52 :crew = 'C'; break;

                           case 53 :crew = 'B'; break;

                           case 54 :crew = 'C'; break;

                           case 55 :crew = 'B'; break;

              }

       }

       LogMessage(DEBUG\_LOG, "Crew = %c", crew);

       return crew;

}

# UC089 – Delays – Slowdown and Delay Report

## Description

This functionality allows to the user to get the Slowdown and Delay Report.

## Triggers

* When the actor clicks on the “Search” button.

## Preconditions

* Actor must select “Slowdown and Delay Report”

## Actors

* User

## Basic Flow

## Basic Flow

## The actor selects the start time and end time to filter the report

## The actor click on the “Search” button

## The system displays the report with the following information (BR1):

## Area

## Description

## Equipment

## Starting heat

## Ending heat

## EAF Stop Time

## Sequence Number

## Time Start

## Time End

## Aim Vc

## Slowest Vc Slow

## Fastest Vc Slow

## Width

## Lost Tons

## Reason.

## End of use case.

## Alternative and Exception Flows

## No alternate or exception flows

## Business Rules

## BR1 – Slowdowns highlight: slowdown rows must be highlighted in grey.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database field** |
| Area | Output | N/A | N/A | N/A | Delays | AREA |
| Description | Output | N/A | N/A | N/A | Delays | DESCRIPTION |
| Equipment | Output | N/A | N/A | N/A | Delays | EQUIPMENT |
| Starting heat | Output | N/A | N/A | N/A | Delays | STARTING\_HEAT |
| Ending heat | Output | N/A | N/A | N/A | Delays | ENDING\_HEAT |
| EAF Stop Time | Output | N/A | N/A | N/A | Delays | EAF\_STOP\_TIME |
| Sequence Number | Output | N/A | N/A | N/A | Delays | SEQUENCE\_NUMBER |
| Time Start | Output | N/A | N/A | N/A | Delays | TIME\_START |
| Time End | Output | N/A | N/A | N/A | Delays | TIME\_END |
| Aim Vc | Output | N/A | N/A | N/A | Delays | AIM\_VC |
| Slowest Vc Slow | Output | N/A | N/A | N/A | Delays | SLOWEST\_VC\_SLOW |
| Fastest Vc Slow | Output | N/A | N/A | N/A | Delays | FASTEST\_VC\_SLOW |
| Width | Output | N/A | N/A | N/A | Delays | WIDTH |
| Lost Tons | Output | N/A | N/A | N/A | Delays | LOST\_TONS |
| Reason | Output | N/A | N/A | N/A | Delays | REASON |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

This report will be a table. No charts.

# UC090 – Mill Production Scheduling – Drag and Drop into Casters List

## Description

This functionality allows to the user to drag a slab from any list and drop it into the Casters List.

## Triggers

* When the actor selects a slab and drag it to the Casters list and drop it there.

## Preconditions

* Actor be in the Mill Production Scheduling screen.

## Actors

* User

## Basic Flow

## The actor selects one or several of the slabs on one of the lists

## The actor drags them to the Casters List and drop them there

## The system updates the slab status to “modified”

## The system set its tundish change and roll change to false

## The system sets its L3 Coil Id, as the next coil id

## The system adds the slab into the list

## End of use case.

## Alternative and Exception Flows

## No alternate or exception flows

## Business Rules

## Not Required.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database field** |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

When the slab its drag and dropped into the Caster list, the following actions must be performed:

* Get the heat id of the slab before the drop
* Set the property bDirty (describes if the slab was modified) to true;
* Set the property ProductionStatus to “” (empty)
* Set the property l3heatid to “” (empty)
* Set the property LastBeforeTundishChange to “false”
* Set the property LastBeforeRollChange to “false”
* Validate if the slab is being dragged from the stock, orderbook, requires disposition or caster list. If its true, then:
  + Assign the L3CoilIdProperty to the next L3 coil id.
* Then, validate if the slab was dropped into the current heat, for each of the slabs on the slabs list, starting by the last position (i--)
  + If the current slab of the cycle, has a productionStatus equal to “C” or “T” or “P”, and the slab is planned type “Normal”, then:
    - Set that the flag for slab found to true
    - Look if the original position of the slab minus 1 is greater than 0. If it is:
      * Get the slab L3 heat id from the original position – 1
      * Validate if the L3 heat Id is the same than the current slab from the cycle. If it is:
        + Set the insertorremoveslabincurrentheat to true
      * Validate if the heat before the drop and the heat id of the current slab on the cycle are the same, if they are:
        + Set the insertorremoveslabincurrentheat to true
    - If the slab found flag is set to true, then break the function.
* Resequence the HSM schedule

**To get the next coil id:**

* Get the value of the next L3 coil id, if its less or equal to 99999 then, return 1 else return the L3 coil id plus 1.

**To resequence the schedule it must perform the following:**

* + Define accumulators for CasterElapsedCastingTime and CasterElapsedStringCasting time, Elapsed Footage, Elapsed Roll Footage, and initialized all to 0
  + Obtain the ORIGINAL quantity of slabs for the schedule, and if its less than 0, set it to 0
  + If the original quantity of slabs is less than the actual quantity on the schedule, obtain the last slab from the list. Then if the index of that last slab is 0, set all accumulator values to 0, if not, assign the following values to the accumulators:
    - Caster Elapsed Casting Time = Elapsed casting time from the slab
    - Caster Elapsed String Casting Time = Elapsed String Casting Time from the slab
    - Elapsed Footage = Coil Length from the slab
    - Elapsed Roll Footage = Elapsed Coil Length
  + Define a new accumulator for the Total Casting time. Check if the slab precedes a roll change. If it does, the Total Casting time accumulator is set to the casting time of the slab, plus the turnaround time. If it does not precedes a roll change, the Total Casting time is set to the casting time of the slab.
  + For each of the slabs additional from the original list (7.5.1.1 BR2, Save all newly created slabs) (ie. If the list had 5 slabs and added 2 more, the additional slabs will be 6 and 7), check if it is in status “I” if its true, assign the following values:
    - Slab Head Cut Time = CasterElapsedCastingTime
    - CasterElapsedCastingTime = CasterElapsedCastingTime + Casting time of the Slab
    - Slab ElapsedCastingTime = CasterElapsedCastingTime
    - Slab ElapsedStringCastingTime = 0.0
    - ElapsedFootage = ElapsedFootage + Slab Coil Length
    - Slab Coil Length = Elapsedfootage
    - If the slab preceedsRollChange, ElapsedRollFootage=slab coil length and SlabTotalCastingtime = turn around time + slab casting time, if not, ElapsedRollFootage= ElapsedRollFootage + slab coil length and SlabTotalCastingTime = slab casting time.
    - Assign to the Mill prod slab sequence the current cycle value plus 1 (i+1)
  + Calculate the cut off points:
    - Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
      * If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
      * Set the slab as cutoff point and end the search for any more cut off points
      * If no cutoff points are located, set it to “NULL”.

# UC091 – Mill Production Scheduling – Drag and Drop into Missed Production

## Description

This functionality allows to the user to drag a slab from any list and drop it into the Missed Production List.

## Triggers

* When the actor selects a slab and drag it to the Missed Production list and drop it there.

## Preconditions

* Actor be in the Mill Production Scheduling screen.

## Actors

* User

## Basic Flow

## The actor selects one or several of the slabs on one of the lists

## The actor drags them to the Missed Production List and drop them there

## The system updates the slab status to “modified”

## The system set its tundish change and roll change to false

## The system sets its L3 Coil Id, as the next coil id

## The system adds the slab into the list

## End of use case.

## Alternative and Exception Flows

## No alternate or exception flows

## Business Rules

## Not required.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database field** |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

When the slab its drag and dropped into the Missed Production list, the following actions must be performed:

* Set the property bDirty (describes if the slab was modified) to true;
* Set the property LastBeforeTundishChange to “false”
* Set the property LastBeforeRollChange to “false”
* Validate if the slab is being dragged from the requires disposition and the L3 coil Id from the slab is empty (“”). If its true, then:
  + Assign the L3CoilIdProperty to the next L3 coil id.
* If its not being dragged from the requires disposition list and the L3 coil id is not empty then:
  + Get the drop index:
    - For each of the slabs on the missed production list box, calidate if the index of the slab being droped is less than the index of the current slab on the cycle, if it is, break the cycle.
* Set the production status of the slab being dragged to “I”.
* Resequence the HSM Schedule
* If the slab is being dragged from the Caster, set the roll change to the previous slab if the roll change slab is dragged.

**To resequence the schedule it must perform the following:**

* + Define accumulators for CasterElapsedCastingTime and CasterElapsedStringCasting time, Elapsed Footage, Elapsed Roll Footage, and initialized all to 0
  + Obtain the ORIGINAL quantity of slabs for the schedule, and if its less than 0, set it to 0
  + If the original quantity of slabs is less than the actual quantity on the schedule, obtain the last slab from the list. Then if the index of that last slab is 0, set all accumulator values to 0, if not, assign the following values to the accumulators:
    - Caster Elapsed Casting Time = Elapsed casting time from the slab
    - Caster Elapsed String Casting Time = Elapsed String Casting Time from the slab
    - Elapsed Footage = Coil Length from the slab
    - Elapsed Roll Footage = Elapsed Coil Length
  + Define a new accumulator for the Total Casting time. Check if the slab precedes a roll change. If it does, the Total Casting time accumulator is set to the casting time of the slab, plus the turnaround time. If it does not precedes a roll change, the Total Casting time is set to the casting time of the slab.
  + For each of the slabs additional from the original list (7.5.1.1 BR2, Save all newly created slabs) (ie. If the list had 5 slabs and added 2 more, the additional slabs will be 6 and 7), check if it is in status “I” if its true, assign the following values:
    - Slab Head Cut Time = CasterElapsedCastingTime
    - CasterElapsedCastingTime = CasterElapsedCastingTime + Casting time of the Slab
    - Slab ElapsedCastingTime = CasterElapsedCastingTime
    - Slab ElapsedStringCastingTime = 0.0
    - ElapsedFootage = ElapsedFootage + Slab Coil Length
    - Slab Coil Length = Elapsedfootage
    - If the slab preceedsRollChange, ElapsedRollFootage=slab coil length and SlabTotalCastingtime = turn around time + slab casting time, if not, ElapsedRollFootage= ElapsedRollFootage + slab coil length and SlabTotalCastingTime = slab casting time.
    - Assign to the Mill prod slab sequence the current cycle value plus 1 (i+1)
  + Calculate the cut off points:
    - Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
      * If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
      * Set the slab as cutoff point and end the search for any more cut off points
      * If no cutoff points are located, set it to “NULL”.

# UC092 – Mill Production Scheduling – Drag and Drop into Stock

## Description

This functionality allows to the user to drag a slab from any list and drop it into the Stock List.

## Triggers

* When the actor selects a slab and drag it to the Stock list and drop it there.

## Preconditions

* Actor be in the Mill Production Scheduling screen.

## Actors

* User

## Basic Flow

## The actor selects one or several of the slabs on one of the lists

## The actor drags them to the Stock List and drop them there

## The system updates the slab status to “modified”

## The system set its tundish change and roll change to false

## The system sets its L3 Coil Id to empty

## The system adds the slab into the list

## End of use case.

## Alternative and Exception Flows

## No alternate or exception flows

## Business Rules

## Not required.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database field** |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

When the slab its drag and dropped into the Stock list, the following actions must be performed:

* Set the property bDirty (describes if the slab was modified) to true;
* Set the property LastBeforeTundishChange to “false”
* Set the property LastBeforeRollChange to “false”
* Set the L3 Coil id to empty (“”)
* Insert into the Order Book List, at the end of the list
* If the slab is being dragged from the caster, then Resequence HSM Schedule

**To resequence the schedule it must perform the following:**

* + Define accumulators for CasterElapsedCastingTime and CasterElapsedStringCasting time, Elapsed Footage, Elapsed Roll Footage, and initialized all to 0
  + Obtain the ORIGINAL quantity of slabs for the schedule, and if its less than 0, set it to 0
  + If the original quantity of slabs is less than the actual quantity on the schedule, obtain the last slab from the list. Then if the index of that last slab is 0, set all accumulator values to 0, if not, assign the following values to the accumulators:
    - Caster Elapsed Casting Time = Elapsed casting time from the slab
    - Caster Elapsed String Casting Time = Elapsed String Casting Time from the slab
    - Elapsed Footage = Coil Length from the slab
    - Elapsed Roll Footage = Elapsed Coil Length
  + Define a new accumulator for the Total Casting time. Check if the slab precedes a roll change. If it does, the Total Casting time accumulator is set to the casting time of the slab, plus the turnaround time. If it does not precedes a roll change, the Total Casting time is set to the casting time of the slab.
  + For each of the slabs additional from the original list (7.5.1.1 BR2, Save all newly created slabs) (ie. If the list had 5 slabs and added 2 more, the additional slabs will be 6 and 7), check if it is in status “I” if its true, assign the following values:
    - Slab Head Cut Time = CasterElapsedCastingTime
    - CasterElapsedCastingTime = CasterElapsedCastingTime + Casting time of the Slab
    - Slab ElapsedCastingTime = CasterElapsedCastingTime
    - Slab ElapsedStringCastingTime = 0.0
    - ElapsedFootage = ElapsedFootage + Slab Coil Length
    - Slab Coil Length = Elapsedfootage
    - If the slab preceedsRollChange, ElapsedRollFootage=slab coil length and SlabTotalCastingtime = turn around time + slab casting time, if not, ElapsedRollFootage= ElapsedRollFootage + slab coil length and SlabTotalCastingTime = slab casting time.
    - Assign to the Mill prod slab sequence the current cycle value plus 1 (i+1)
  + Calculate the cut off points:
    - Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
      * If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
      * Set the slab as cutoff point and end the search for any more cut off points
      * If no cutoff points are located, set it to “NULL”.

# UC093 – Mill Production Scheduling – Drag and Drop into Required Disposition

## Description

This functionality allows to the user to drag a slab from any list and drop it into the Requires Disposition List.

## Triggers

* When the actor selects a slab and drag it to the Requires Disposition list and drop it there.

## Preconditions

* Actor be in the Mill Production Scheduling screen.

## Actors

* User

## Basic Flow

## The actor selects one or several of the slabs on one of the lists

## The actor drags them to the Requires Disposition List and drop them there

## The system updates the slab status to “modified”

## The system set its tundish change and roll change to false

## The system sets its L3 Coil Id to empty

## The system adds the slab into the list

## The system checks if the inserted slab has low CU.

## End of use case.

## Alternative and Exception Flows

## No alternate or exception flows

## Business Rules

## Not required.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database field** |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

When the slab its drag and dropped into the Requires Disposition, the following actions must be performed:

* Set the property bDirty (describes if the slab was modified) to true;
* Set the property LastBeforeTundishChange to “false”
* Set the property LastBeforeRollChange to “false”
* Validate for each of the slabs:
  + If the order of the slab on the cycle is greater than the order of the dropped slab, or If the order of the slab on the cycle is equal than the order of the dropped slab and the line item number is greater than the line item number of the dropped slab, or If the order of the slab on the cycle is equal than the order of the dropped slab and the line item number is equal to the line item number of the dropped slab and the line item coil number is also equal to the line item coil number of the dropped slab, then break the cycle.
* Calculate the Sums of the dropped slab:
  + Calculated weight produced = SELECT ISNULL(SUM(coil\_product\_data.actual\_coil\_weight),0) FROM coil\_product\_data ,coil WHERE coil.production\_coil\_no = coil\_product\_data.production\_coil\_no AND coil.production\_coil\_no IS NOT NULL AND coil.order\_no = [DRAGGED SLAB ORDER NO] AND coil.line\_item\_no =[DRAGGED SLAB LINE ITEM NO]
  + Calculated weight to be produced= SELECT ISNULL(SUM(coil.mill\_weight\_aim),0) FROM coil WHERE coil.coil\_status in ('CO','IS') AND coil.order\_no = [DRAGGED SLAB ORDER NO] AND coil.line\_item\_no ==[DRAGGED SLAB LINE ITEM NO]
* Insert the slab into the Requires disposition list
* Set the L3 Coil Id to empty
* Check for the low CU property, if it is yes, paint the background in cupper color.
* Validate if the list of origin of the dragged slab is from the Caster, if it does, Resequence the HSM Schedule

**To resequence the schedule it must perform the following:**

* + Define accumulators for CasterElapsedCastingTime and CasterElapsedStringCasting time, Elapsed Footage, Elapsed Roll Footage, and initialized all to 0
  + Obtain the ORIGINAL quantity of slabs for the schedule, and if its less than 0, set it to 0
  + If the original quantity of slabs is less than the actual quantity on the schedule, obtain the last slab from the list. Then if the index of that last slab is 0, set all accumulator values to 0, if not, assign the following values to the accumulators:
    - Caster Elapsed Casting Time = Elapsed casting time from the slab
    - Caster Elapsed String Casting Time = Elapsed String Casting Time from the slab
    - Elapsed Footage = Coil Length from the slab
    - Elapsed Roll Footage = Elapsed Coil Length
  + Define a new accumulator for the Total Casting time. Check if the slab precedes a roll change. If it does, the Total Casting time accumulator is set to the casting time of the slab, plus the turnaround time. If it does not precedes a roll change, the Total Casting time is set to the casting time of the slab.
  + For each of the slabs additional from the original list (7.5.1.1 BR2, Save all newly created slabs) (ie. If the list had 5 slabs and added 2 more, the additional slabs will be 6 and 7), check if it is in status “I” if its true, assign the following values:
    - Slab Head Cut Time = CasterElapsedCastingTime
    - CasterElapsedCastingTime = CasterElapsedCastingTime + Casting time of the Slab
    - Slab ElapsedCastingTime = CasterElapsedCastingTime
    - Slab ElapsedStringCastingTime = 0.0
    - ElapsedFootage = ElapsedFootage + Slab Coil Length
    - Slab Coil Length = Elapsedfootage
    - If the slab preceedsRollChange, ElapsedRollFootage=slab coil length and SlabTotalCastingtime = turn around time + slab casting time, if not, ElapsedRollFootage= ElapsedRollFootage + slab coil length and SlabTotalCastingTime = slab casting time.
    - Assign to the Mill prod slab sequence the current cycle value plus 1 (i+1)
  + Calculate the cut off points:
    - Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
      * If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
      * Set the slab as cutoff point and end the search for any more cut off points
      * If no cutoff points are located, set it to “NULL”.

# UC094 – Mill Production Scheduling – Drag and Drop into Workspace

## Description

This functionality allows to the user to drag a slab from any list and drop it into the Workspace List.

## Triggers

* When the actor selects a slab and drag it to the Workspace list and drop it there.

## Preconditions

* Actor be in the Mill Production Scheduling screen.

## Actors

* User

## Basic Flow

## The actor selects one or several of the slabs on one of the lists

## The actor drags them to the Workspace List and drop them there

## The system updates the slab status to “modified”

## The system set its tundish change and roll change to false

## The system sets its L3 Coil Id.

## The system adds the slab into the list

## End of use case.

## Alternative and Exception Flows

## No alternate or exception flows

## Business Rules

## Not required.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database field** |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

When the slab its drag and dropped into the Workspace, the following actions must be performed:

* Validate if the slab comes from the stock, orderbook, or requires disposition lists, and that the L3CoilId is equal to 0. If it is:
  + Set the L3CoilId of the dragged slab to the next l3 coil id.
* Set the property bDirty (describes if the slab was modified) to true;
* Set the property LastBeforeTundishChange to “false”
* Set the property LastBeforeRollChange to “false”
* Insert the slab into the workspace list
* If the slab is dragged from the caster, set the roll change to previous slab, if the slab being dragged is the roll change slab
* If the slab is dragged from the caster or the missed production, Resequence the HSM Schedule

**To resequence the schedule it must perform the following:**

* + Define accumulators for CasterElapsedCastingTime and CasterElapsedStringCasting time, Elapsed Footage, Elapsed Roll Footage, and initialized all to 0
  + Obtain the ORIGINAL quantity of slabs for the schedule, and if its less than 0, set it to 0
  + If the original quantity of slabs is less than the actual quantity on the schedule, obtain the last slab from the list. Then if the index of that last slab is 0, set all accumulator values to 0, if not, assign the following values to the accumulators:
    - Caster Elapsed Casting Time = Elapsed casting time from the slab
    - Caster Elapsed String Casting Time = Elapsed String Casting Time from the slab
    - Elapsed Footage = Coil Length from the slab
    - Elapsed Roll Footage = Elapsed Coil Length
  + Define a new accumulator for the Total Casting time. Check if the slab precedes a roll change. If it does, the Total Casting time accumulator is set to the casting time of the slab, plus the turnaround time. If it does not precedes a roll change, the Total Casting time is set to the casting time of the slab.
  + For each of the slabs additional from the original list (7.5.1.1 BR2, Save all newly created slabs) (ie. If the list had 5 slabs and added 2 more, the additional slabs will be 6 and 7), check if it is in status “I” if its true, assign the following values:
    - Slab Head Cut Time = CasterElapsedCastingTime
    - CasterElapsedCastingTime = CasterElapsedCastingTime + Casting time of the Slab
    - Slab ElapsedCastingTime = CasterElapsedCastingTime
    - Slab ElapsedStringCastingTime = 0.0
    - ElapsedFootage = ElapsedFootage + Slab Coil Length
    - Slab Coil Length = Elapsedfootage
    - If the slab preceedsRollChange, ElapsedRollFootage=slab coil length and SlabTotalCastingtime = turn around time + slab casting time, if not, ElapsedRollFootage= ElapsedRollFootage + slab coil length and SlabTotalCastingTime = slab casting time.
    - Assign to the Mill prod slab sequence the current cycle value plus 1 (i+1)
  + Calculate the cut off points:
    - Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
      * If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
      * Set the slab as cutoff point and end the search for any more cut off points
      * If no cutoff points are located, set it to “NULL”.

# UC095 – Mill Production Scheduling – Drag and Drop into OrderBook

## Description

This functionality allows to the user to drag a slab from any list and drop it into the OrderBook List.

## Triggers

* When the actor selects a slab and drag it to the OrderBook list and drop it there.

## Preconditions

* Actor be in the Mill Production Scheduling screen.

## Actors

* User

## Basic Flow

## The actor selects one or several of the slabs on one of the lists

## The actor drags them to the OrderBook List and drop them there

## The system updates the slab status to “modified”

## The system set its tundish change and roll change to false

## The system sets its L3 Coil Id to empty

## The system adds the slab into the list

## End of use case.

## Alternative and Exception Flows

## No alternate or exception flows

## Business Rules

## Not required.

## Input / Output Fields Properties and Validations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen/Field** | **Type** | **Required** | **Default Value** | **Validations / Rules** | **Database Table** | **Database field** |
|  |  |  |  |  |  |  |

## Custom MII Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Sample Data** | **Rules / Comments** |

## Custom Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Required** | **Type** | **Validation** |
|  |  |  |  |  |  |

## Flow Chart

None.

## Technical Specification

When the slab its drag and dropped into the Stock list, the following actions must be performed:

* Set the property bDirty (describes if the slab was modified) to true;
* Set the property LastBeforeTundishChange to “false”
* Set the property LastBeforeRollChange to “false”
* Set the L3 Coil id to empty (“”)
* Insert into the Order Book List, at the end of the list
* If the slab is being dragged from the caster, then Resequence HSM Schedule

**To resequence the schedule it must perform the following:**

* + Define accumulators for CasterElapsedCastingTime and CasterElapsedStringCasting time, Elapsed Footage, Elapsed Roll Footage, and initialized all to 0
  + Obtain the ORIGINAL quantity of slabs for the schedule, and if its less than 0, set it to 0
  + If the original quantity of slabs is less than the actual quantity on the schedule, obtain the last slab from the list. Then if the index of that last slab is 0, set all accumulator values to 0, if not, assign the following values to the accumulators:
    - Caster Elapsed Casting Time = Elapsed casting time from the slab
    - Caster Elapsed String Casting Time = Elapsed String Casting Time from the slab
    - Elapsed Footage = Coil Length from the slab
    - Elapsed Roll Footage = Elapsed Coil Length
  + Define a new accumulator for the Total Casting time. Check if the slab precedes a roll change. If it does, the Total Casting time accumulator is set to the casting time of the slab, plus the turnaround time. If it does not precedes a roll change, the Total Casting time is set to the casting time of the slab.
  + For each of the slabs additional from the original list (7.5.1.1 BR2, Save all newly created slabs) (ie. If the list had 5 slabs and added 2 more, the additional slabs will be 6 and 7), check if it is in status “I” if its true, assign the following values:
    - Slab Head Cut Time = CasterElapsedCastingTime
    - CasterElapsedCastingTime = CasterElapsedCastingTime + Casting time of the Slab
    - Slab ElapsedCastingTime = CasterElapsedCastingTime
    - Slab ElapsedStringCastingTime = 0.0
    - ElapsedFootage = ElapsedFootage + Slab Coil Length
    - Slab Coil Length = Elapsedfootage
    - If the slab preceedsRollChange, ElapsedRollFootage=slab coil length and SlabTotalCastingtime = turn around time + slab casting time, if not, ElapsedRollFootage= ElapsedRollFootage + slab coil length and SlabTotalCastingTime = slab casting time.
    - Assign to the Mill prod slab sequence the current cycle value plus 1 (i+1)
  + Calculate the cut off points:
    - Define a new castercutoffpoint variable and Go through each of the slabs on the active schedule, and for each of them, validate as follows:
      * If the slab status is not “I” and is equal to blank or “P”, then set the CasterCutOffPointVariable as the current slab position (i)
      * Set the slab as cutoff point and end the search for any more cut off points
      * If no cutoff points are located, set it to “NULL”.