# F152832

## Ensure the user will see a status field

### Where should the new field called status be?

### should the user be able to move or hide this field?

### What does R status mean?

[‎7/‎1/‎2019 2:31 PM]  Rose, Bob (A.):

Good afternoon Benjamin,  
I'm working on F152832 Build to WERS - In CMF WERS Notice Tab need status of notice going to R status   
I'm in the middle of acclimating to the first couple of Features from Chris which lead me to a nice search results of over 90 related Features.  I came across F50863 wile looking for some clarification on Notice status "R"  
What does Notice "R" mean?  
I first assumed it meant Rejected but when I studied the history and talked to a Trainer, we never see a CMF process approved in Rejected Status.  History shows only user created CMFs are Rejected.  
  
What does Notice "R" mean?

[‎7/‎1/‎2019 2:34 PM]  Pohl, Benjamin (B.):

R status means released

### What does C status mean?

#### C mean completed

## Questions for the Customer

### What are the expectations

#### WERS Notices appear to be issues

#### Are you wanting a report on Released parts that had an issue?

#### How many occurrances are you expecting

#### Is there any particular program

## Rally data integrity

### Which related features have test cases?

## What is the procedure to get the WERS notice to show data

### F34405

#### Team Center must validate replaces/replaced by at time of authoring and certification

#### PPM should not receive any replaces/replaced by build errors

#### Current Build Errors

##### INVALID REPLACED BY DATA ON WERS SLOC FUNC

###### (#9)

##### FEDEBOM fix

###### NO MATCHING REPLACES PART EFFECTIVE OUT

(#4, #3)

##### FEDEBOM fix (WERS fix if activity is different)

###### NO REPLACED BY PART EFFECTIVE IN FOUND

(#1, #4)

##### FEDEBOM fix

###### NO REPLACES PART USAGE FOUND

(#1, #4)

##### FEDEBOM fix

###### REPLACE-BY PART –

###### INVALID CPSC –

##### FEDEBOM fix

###### REPLACES AND REPLACED-BY PARTS ARE THE SAME (#12, #5)

##### FEDEBOM fixREPLACES PART

##### INVALID CPSC

##### FEDEBOM fixREPLACES PART

##### INVALID PART FORMAT

##### FEDEBOM fixREPLACES PART FUNCTION DOES NOT CONTAIN ITS CORRESPONDING REPLACED BY PART

##### FEDEBOM fix (WERS fix if activity is different)REPLACES/REPLACED BY NOT FOUND

##### FEDEBOM fixR/RP PAIR DOES NOT BUILD DIFFERENT ACTIVITIES (#15/16)

##### WERS and FEDEBOM fixPART ALREADY EXISTS AS A SINGLE OR NON-COLOR PARTPART ALREADY EXISTS AS AN IN-WHITE PART

###### Rules to follow

###### 1. Team Center must validate any replaces/replaced by part information (the replaces or replaced by part must exist on the BoM)

### Test Runs

#### How are auto build end items identified?

##### A recent change allows this to be set at an end item level

###### In the BOM Config form

Set auto build on for an effective point

What role is needed to access this?

###### Also impacts components

##### By program

# F10324 WERS - Assign Sequence Number for AVBOM Authored LoU's (FRCC)

* When a new usage is added by an End User in the Wizard/CMF and passes through the build process into WERS, the correct sequence number will be determined by WERS, applied to the usage and passed back to BOMF when required.  See attached file for the Split-Worthy fields.
  + Actors:
    - Primary: D&R Engineer
    - Secondary: BOMF
  + Steps:
    - An End User (D&R Engineer, CMF Approver or BOM Admin) will author a new line of usage (LoU) in the Wizard for a Build On program with a default Sequence value of 000
    - After submitting the changes the usage will be passed to CMF
    - The CMF will either be system or manually approved and the changes will be sent to BOMF
    - WERS will be listening for Build On program updates in BOMF (via Pub/Sub environment) and write the changes into WERS
    - Prior to taking the Notice to R-Status, WERS will compare the Usage against the FRCC table in WERS to determine the function record in WERS that aligns to the AVBOM/BOMF usage
    - If that function record has a Sequence value that is different than what was provided by BOMF, WERS will send the correct sequence value back to BOMF and it will be stored against that usage
  + Acceptance Criteria:
    - Given <an End User adding a new LoU in the Wizard> when <the End User is entering all required part and usage attribute information> then <the Wizard will display the Sequence value as 'TBD' and it will be read-only>
    - Given <an End User working in the Wizard> when <the End User hovers the cursor over the Sequence header cell> then <the Wizard will display an updated definition of Sequence that explains the new changes in behavior>
    - Given <an End User adding a new LoU in the Wizard> when <the End User is entering all required part and usage attribute information> then <the Wizard will default the Sequence value to '000' in the background and send that value in the submit XML>
    - Given <a BOM Admin adding a new usage via admin import> when <the BOM Admin submits the import file> then <Sequence will not be a required field by BPM to create the LoU>
    - Given <a new LoU being added via AVBOM> when <the LoU is passed to WERS to be built> then <WERS will analyze the key usage attributes and split-worthy fields to determine the corresponding function record>
    - Given <a new LoU being added via AVBOM> when <an existing function record aligns to the key usage attributes and split-worthy fields> then <WERS will attach that function record to the notice and send the Sequence value back to BOMF (if different than what was received from BOMF)>
    - Given <a new LoU being added via AVBOM> when <no existing function record aligns to the key usage attributes and split-worthy fields> then <WERS will create a new function record, assign the lowest available sequence value (starting with 000) and pass that value back to BOMF (if different than what was received from BOMF)>

# **F10397 WERS - Bypass CAT III Validations for Auto-Build Notices**

When an auto-build notice is sent from CMF to WERS, WERS should bypass the existing CAT III validations and allow the Notice to be built and taken to Complete status.

Actors:

* Primary: CMF
* Secondary:

Steps:

* For Build On programs an auto-build notice will be generated in WERS
* WERS will build the notice and bypass the existing CAT III validations, allowing any CAT III parts, functions and usages to be released

Acceptance Criteria:

* Given <an auto-build notice from CMF> when <the Notice includes a base number in the CAT III (AAJT) table> then <WERS will ignore the existing CAT III validations and allow the notice to be built>
* Given <a manually authored notice in WERS> when <the Notice includes a base number in the CAT III (AAJT) table> then <WERS will run the existing CAT III validations>

Assumptions:

* No changes are required for Validation

# F10416 BOMF - Promote BOM work to published status

Note:  This feature has been split.  Remaining work to complete Publish is in [F14003](https://rally1.rallydev.com/#/34494673187d/detail/portfolioitem/feature/40120148421).

Actor:  BOMF client

Trigger:  When called by a client (e.g. CMF) to promote a set of frozen BOM records to published status.

Steps:

1)  Create transaction record (in G22) with Change Notice number provided by CMF

2)  Create TRANS\_LOU records (G42) for the LOUs CMF asked to publish

3)  Start (Oracle) transaction

3a)  Obtain a lock (select for update) on all parts which will be involved in the publish operation - code will wait here until all locks can be obtained  
4) Across all LOUs to be published, for all frozen records for a given target table (in G43), for each group of those records which refer to the same unique key (row) in that target table (sort and process that group in the order in which the records were frozen):

  a)  Happy path - if there is only one frozen record in that group, and if the frozen record's origin record is (still) published, then effect-out the the current published record (and set it to history status) and effect-in the frozen record (and set it to published status)

  b)  Else (merge path):

    1.  Create a working copy of the current published record as a starting point for the new record to be inserted and published

     2.  For each record in the group

a.  Compare the record with its' origin record

b.  Determine which attributes are different (i.e. which ones the author was trying to change)

c.  Apply those changed attributes to the working copy of the new record to be published (note that if the attribute is a cost/weight specified as a "delta", then apply the *difference* (between the frozen value and the origin value for that attribute) to whatever is currently in the new "to be published" record for that attribute).  For example, if the frozen record has a weight value of 10 flagged as a delta, and its' origin record has a value of 2, then add 8 (the difference) to whatever is in the "to be published" record at this point in the algorithm (for example 6) so the new weight value to be published would be 14.

     3.  Effect-out the currently published record and set it to H status

     4.  Insert the "to be published" record, effect-in as of "now", and set it to P status.  Get sakey of record just inserted for next step.

     5.  Set the status of all the frozen records we included in the merge to M (merged) status, and update them to point to the record we just published, that they were merged into

Note  For traceability purposes, we might want to insert records into G43 to indicate which tables we updated as part of publishing each LOU.  We definitely want a record in G42 for the record we set to P status.  It might be desireable to have records in G43 for the record we set to H status and any records we set to M status, but those records aren't strictly necessary

3) In case of error (including timeout while attempting to obtain a lock on the parts), capture the error information, the overall transaction should show "failed" in G22, and the records in G42 should remain (showing what LOUs we were trying to publish).  Rollback changes to all other tables (including releasing locks on part table) return failed status to the caller (e.g. CMF).

3a) Commit Oracle transaction and release locks on Parts

The following two steps have been moved to a separate "Push to WERS" feature

4)  Within a DB2 transaction, save all published LOUs in DB2 for Validate and trigger WERS via MQ to perform the Validate

5)  In case of error, rollback and exit, otherwise commit

A separate feature has been created to create the means to invoke post-processing with a reflection CO, but it will need to be called from here as part of the "Publish" operation.

6)  Asynchronously, invoke BPM post-processing

Acceptance Criteria:

1)  GIVEN <a set of frozen BOM data in BOMF>, WHEN <told to publish the frozen BOM data for a set of LOU IDs> THEN <BOMF publishes that data, performing merge processing if necessary>

2)  GIVEN <merge is required> WHEN <the Frozen record represents a delta from its ancestor record> THEN <BOMF should subtract the ancestor attribute value from the Frozen value to determine the amount of the delta, then apply that amount to the current published attribute>

3)  GIVEN <a set of BOM data has been published> WHEN <finished publishing> BOMF will invoke BPM to perform post-processing on that change>

Notes:

1)  As part of publishing, BOMF may need to coalesce/aggregate multiple changes to same object within same CMF (created by potentially different users, at different points-in-time, potentially with different ancestry, which came-in as part of different bundles).  In other words, two different Wizard sessions might modify the same or different attributes in the same petal table.  Both changes would be held (as we understand it) as different changed LOUs in CMF.  On publishing, BOMF will merge those changes to the same petal table row into a single new  published record.

2)  The input parameter is a list of LOU IDs to be published.   **The unit of work is the entire set of LOUs.**

3)  How does BOMF know the frozen record's origin?  By "origin" attributes in the frozen records.  See sub-epic [SE2503](https://rally1.rallydev.com/#/34494673187d/detail/portfolioitem/subepic/35939434609)

4)  Atamjeet suggested that we can replace the existing semaphore mechanism (which requires inserting and deleting records from a semaphore table) by getting “select for update” locks on Parts in the Part table instead.  This will keep different “publish” transactions (which would be occurring at the process-approved CMF or change notice level) from intermixing updates to the same part or its associated usage/cost data.  This will leverage in-memory database locking which should be much less overhead.  If this turns-out to be insufficient, we can implement something more robust (i.e. a semaphore table which we will poll every so often, with some set maximum time before we give up and declare a publish failure).

5)  When promoting to published, the call from CMF will include a Change Notice number.

6)  Per discussion with Markus, we should first evaluate if *any* of the frozen records being published for a LOU will require a merge.  If so, we should proceed into a code path to handle the entire LOU update as a merge.  In our initial implementation of that "LOU merge", we will still go ahead and flip the records to published that can be flipped, and merge the records that need to be merged, across the various tables that need to be updated for that LOU.  In the future, we may want to e.g. notify the author that a merge is required on the LOU he changed, and does he want to go ahead with publishing that LOU or any part of it. For the purposes of this feature, won't notify anyone, and won't consider context.

7)  "Push to WERS" and "Invoke Post Processing" features have been split-off from this feature into separate features.  The publish code will still need to call the feature to invoke post-processing.  CMF will call the BOMF "Push to WERS" feature (for validate or build) separately, after the "publish" call to BOMF successfully returns.

# F10422 CMF - Create Change Notices

Trigger:  CMF Approved, build on for program / effective point

* Actors:
  + Primary: Change Approver
  + Secondary: WERS
* Steps:
  + Change is approved in CMF
  + CMF interacts with WERS to (re) validate data
  + CMF creates change notices
* Acceptance Criteria:
  + Given <a business-approved CMF> when <approved> then <the BOM data is associated with a Change Notice in CMF>

# F105699 TC2WERS: PCA Update Logic for P Authority LLF Build (revising F57830)

# **F10719 CMF - Promote BOM work to published status NA**

Trigger:  User approves CMF (or CMF is process-approved, or CMF is off) and BOM data has been Saved and Frozen in BOMF

Actor:  D&R Engineer or CMF user

Steps:

1)  Actor approves change (or change has been saved/frozen, and didn't require business approval)

2)  CMF calls BOMF, passing LOU IDs to be published and Change Notice Number

3)  BOMF publishes the data

4)  CMF calls BOMF, passing Change Notice Number and Build/Validate flag

5)  BOMF pushes the LOUs for that CN to WERS for build/validate as appropriate

6)  CMF calls WERS to build the notice (there is another feature [F8591](https://rally1.rallydev.com/#/34494673187d/detail/portfolioitem/feature/34497219229) for this)

Acceptance Criteria:

1)  GIVEN there is BOM work in Frozen state in BOMF, WHEN CMF is ready to publish BOM work, THEN CMF calls BOMF to publish a specific set of LOUs (including a Change Notice)

2)  GIVEN there is BOM work in Frozen state in BOMF, WHEN errors occur during publishing  THEN CMF logs the errors and notifies Business Support via email.

Notes:

1)  CMF splits the LOUs for a given CMF into those which are build-on *and* PPM-certified, and those which are not.

* + The LOUs which are build-on and PPM-certified are chipped into CNs based on activity & base
  + The other LOUs are associated with one or more CNs (i.e. they may be chipped by activity/base if it makes the code much simpler, but at a minimum they are assigned a single CN)

# F10748 WERS - Create Design Records

* When an auto-build notice is created in WERS and it includes a new part number that is not yet released in WERS, WERS will create a new design record for the part using the information in the notice.
  + Actors:
    - Primary:  CMF (Request to build comes from CMF)
    - Secondary:  BOMF (has previously stored the PartType XML for the LOUs to be built in WERS DB2 table)
  + Steps:
    - WERS will receive the BOM change information from BOMF and CMF
    - WERS will create the auto-build notice
    - WERS will create a new design record using the information from BOMF
  + Acceptance Criteria:
    - Given <an auto-build notice in WERS> when <the notice includes a new part that is not released in WERS> then <WERS will automatically create a design record for the new part

# F11121 WERS - Change Local Notice Group (LNG) to Change Notice (CN)

As part of preparation for AFSU, some concerns have been identified with the Local Notice Group concept:

LNG is not a business-recognized term.  We have agreed that a better term for this concept is "Change Notice" - which corresponds 1:1 to "WERS Notices" (you could say that WERS Notices are a type of Change Notice, or that both are a type of Change Order).

Therefore the desired changes are:

1)  Any screens that display something like "LNG #" or "Local Notice (Group) #" must be changed to display "Change Notice #" instead

2)  We would prefer, if the effort is not too significant, to rename tables and columns from "LNGxxx" or "Local Noticexxx" to "Change Notice", and also to rename such concepts where they appear in the code (currently in CMF and WERS) e.g. instead LocalNoticeGroupBO we would have ChangeNoticeBO, and instead of talking about generating Local Notices / Local Notice Groups, the code should refer to generating Change Notices instead.

3)  The concatenated Local Notice Group Number field should be removed from the ...Notice table

Business Value:

1)  Use business terms to refer to these entities

2)  Align the code and database to these business terms and improve maintainability

3)  Removal of unnecessary code to create and manipulate unnecessary fields

# F11316 WERS - Publish Notice Processing Notification to Subscribers

Trigger: WERS receives Change Notice from CMF

* Actors:
  + Primary:     CMF
  + Secondary: BOMF
* Pre-conditions:
  + CMF has created a CN
  + Upon  BOMF publish for Build on LOUs, BOMF has written to WERS DB2 transactional
  + Upon successful save of entire CN, CMF has published CN and its metadata
  + Systems (e.g. CMF/BOMF) have been configured to subscribe to WERS build notifications
  + WERS has built associated Design, Function and usage (from DB2 transactional)
* Steps:
  + WERS publishes results of notice processing
* Acceptance Criteria:
  + A notice is created that reflects the CN
  + All Designs, Function and usage records for a CN are built in WERS

# F114568 D&R selects 'Submit to CAD' in Initial P CMF by CMF Notice

**Description/Business Objective:** User has to select 'Submit to CAD' button to generate a WERS Notice (F120276, F120422) and send information to CAD (F114570) to progress workflow.

Once 'Submit to CAD' button is selected for a CMF notice:

1. The Status should change from 'Not Submitted to CAD' to 'WERS Notice being created'
2. WERS Notice should be populated in 'CMF Notice Details' Screen' and status changed to 'CAD in Progress'

**Business Capability:**2.6.2.1.2 Bill Of Material Attribute Management

**Problem Statement:** Need a way to tell CAD the grouping is complete (as of right now) and they can start progress the TeamCenter workflow.

**Risk:** To be filled out with the IT architect team during the 'discovery' phase

**SME needed:**  Muthu

**Acceptance Criteria:**

1. GIVEN <a BOM user is working on an initial P CMF> WHEN <Parts are grouped correctly in the CMF Notices> THEN <the BOM user can press submit to CAD button> AND <status should change from 'Not submitted to CAD' to 'WERS Notice being created'>
2. GIVEN <the status on an initial P CMF is 'WERS Notice being created> WHEN <Notice information has been sent back to FEDE BOM> THEN <WERS Notice should be populated on the CMF Notice Details Screen> AND <status should change to 'CAD in Progress'>

# F114570 End Item and Component Information Sent to TeamCenter after Initial P-Release Grouping

**Description/Business Objective:** After a D&R selects 'Submit to CAD button' and a WERS notice is generated, the required information needs to be sent to TeamCenter via a CSV file in order for them to start their workflow the parts to support Initial P release ahead of FDJ.  
  
CSV file mock-up attached  
  
IT to work with TeamCenter team on where to post file.

**Business Capability:**2.6.2.1.2 Bill Of Material Attribute Management

**Problem Statement:** Need a way to communicate BOM information to CAD in order for them to start their workflow for Initial P release.

**Risk:** To be filled out with the IT architect team during the 'discovery' phase

**SME needed:** **Derek Onkka (DONKKA2) covering feature from 4/29 - 5/3**

**Acceptance Criteria:**   
GIVEN <A D&R raises an Initial P CMF in FEDE BOM>  
WHEN <they select 'Sumit to CAD' for a CMF Notice>  
AND < A WERS Notice placeholder is generated>  
THEN <a CSV file should be generated from FEDE BOM and posted to a shared location for TeamCenter to consume>

# F114572 Display Status against End Items within a Initial P Release CMF

**Description/Business Objective:**Need a way to view CMF notice and CAD status at a CMF notice level and End Item level in the CMF tool for Initial P-release.

**Valid Status' and Descriptions:**

* Not Submitted to CAD - End user has **not** selected 'Submit to CAD' for the CMF Notice Grouping
* WERS Notice being Created - End user has selected 'Submit to CAD' for the CMF Notice Grouping and system is auto-generating notice
* CAD in Progress -  WERS notice is generated and in CMF notice, sent to CAD, and CAD is **not** complete
* CAD Complete - End user has selected 'Submit to CAD' for the CMF Notice Grouping and CAD is complete
* Final Approved - CMF Notice Grouping has been Final Approved
* Error - Error in connection between systems

If one or many EIs have CAD in Progress, the CMF notice status should remain 'CAD in Progress' --- only change to 'CAD Complete' when all End Items within a CMF notice have a status of 'CAD Complete'

If a CMF notice grouping is changed - parts added / deleted... reset the status to 'Not Submitted to CAD' and require 'Submit to CAD' again.

**Business Capability:**2.6.2.1.2 Bill Of Material Attribute Management

**Problem Statement:** Currently do not have functionality in CMF to view CAD status - this is required for initial P release and will be required for Post FDJ Change Control.

**Risk:** To be filled out with the IT architect team during the 'discovery' phase

**SME needed:** Muthu

**Acceptance Criteria:** Needed to reach 'defining' state

1. GIVEN <a CMF Notice is generated> WHEN <the End User Submits to CAD> THEN <the Status Should change from 'Not Submitted to CAD' to 'WERS Notice being Created' for the CMF Notice (and all EI's)>
2. GIVEN <a CMF Notice was sent to CAD> WHEN <FEDE BOM recieves a notice number from WERS> THEN <the Status Should change from 'WERS Notice being Created' to 'CAD in Progress' for the CMF Notice (and all EI's)>
3. GIVEN <a CMF Notice was sent to CAD> WHEN <FEDE BOM recieves a notification from TeamCenter that CAD is complete on an End Item with in CMF Notice> THEN <the Status Should change from 'CAD in Progress' to 'CAD Complete' for that one End Item> AND <The status for CMF Notice should remain 'CAD In Progress>
4. GIVEN <a CMF Notice was sent to CAD> WHEN <FEDE BOM recieves a notification from TeamCenter that CAD is complete on all end items with a CMF Notice> THEN <the Status Should change from 'CAD in Progress' to 'CAD Complete' for the CMF Notice>
5. GIVEN <a CMF Notice has a status of 'CAD Complete'> WHEN <the Approver Final Approves the CMF Notice> THEN <the Status Should change from 'CAD Complete' to 'Final Approved' for the CMF Notice>

# Build to WERS - In CMF WERS Notice Tab need status of notice going to R status

# Train on how to perform this procedure

## when will the training site be updated

# identify data conditions needed to execute this procedure

# test this procedure in the lower environment and take screen shots

# TEST RUN

## Part SLOC = FEDEBOM

## WERS Release populated

## Pbs E4B5 A23942 APW

## Replace with

## Pbs E4B5 A23942 BBW PNL ASY FRT DR TR

## Errors prevented this action

# Approach Research

## Use the largest bom B562\_DU to find parts to add or replace

## Use a smaller bom p558 to make the adjustments

## Add an WPR ASY W/SHLD E4B5 17500 AE

# mark up screen shots

# build a matrix of the security model

# build UML diagram of the software

# build a test plan

# build test cases

# obtain UAT

# Identify dependancies

# code dependancies

# create flow of the process with screen shots for each path that can be taken

# create list of all attributes involved before and the expectation of after

# create all call stacks

# create list of all variables active before and after forms seen by user

# create software flow going back to where each variable was initiated and changed for all paths

# Assumptions

# FEDEBOM will pass WERS a valid WERS notice activity

# The maximum notice supplements will be no more than 999 per CMF (Will they go over 999? No, per Jessica)

# out of scope

# history of status changes

# does R status mean reviewed or rejected

# There are 94 potentially Relevant Rally Items

# F114568 D&R selects 'Submit to CAD' in Initial P CMF by CMF Notice

# F120276 with ppt attached

# F120422 Generate Concern and Notice Placeholders Prior to Final Approval for Initial P CMF Notices (FEDE BOM Team)

# F141330 Auto-Build to detect Notice Groupings being deleted and Delete WERS Notice

# F114572

# list the WERS Notices found in the Features and stories

# attempt to create cmf's for each

# list the related test cases

# find cmf's with WERS Notice by filtering on cmf status Complete and Build Error(scroll down to)

# cmd 1066307

# cmf 1066300