**This document is intended to be viewed alongside the envisioned Test Configuration here** - <https://tenasys.sharepoint.com/:w:/r/sites/Engineeringresources/Shared%20Documents/Testing/Test%20configuration.docx?d=w371d0f4ee12941edb627a7793f28faba&csf=1&web=1&e=eWSe0Y>

**Terminology:**

*Web config* – Current mechanism of configuration machine/tests (<http://lab/config.html)>

*Input config* – *“new approach”* – *S*ome other mechanism of providing the input for machine/test configuration (CSVs or relational database)

*Hosts page* – <http://lab/>

*Results page* - http://lab/results.html

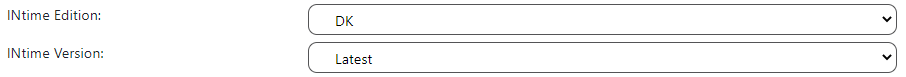
**Current Test Configuration:**

* Items 1,2 currently done by choosing a lab machine that fits current needs via [Hosts page.](http://lab/)
* Items 3,4 done via web config (INtime edition, INtime Version).
* Item 5 – no “preset” configuration, it is as tests require. e.g. If a test specifies “nodeb” then we will add it prior to running test if it does not already exist.
  + Adding nodes
  + Node memory
  + Sets whether to use node memory above 4gb
* “Test Suites” (Item 6) – written in a testing framework such as CUNIT or PYTEST (CPP/Boost are custom)
  + A “test suite” is ran via a Chef recipe (execute\_ifw\_clibrary\_tests).
  + A “test group” is a Chef role (tests\_ifw\_sdk).
* “Test Configurations” (Item 7) - will adjust settings such as
  + Xm/noxm
  + Where to launch tests (nodeb)
* Tests/Test Groups – result(s) in TestRail with title and metadata about the environment the test(s) were run in. By default, and with few exceptions, a “test result” will correlate to a test result file. This means, executing clibrary tests in xm mode and then in non-xm mode will be two result files (we call the CUNIT test cases twice via Piperta) thus two TestRail results. We can investigate merging results, but we would need to flush out what that would look like to get an idea of effort. There’s complexity such as knowing when all results that need to be merged are complete.

**Changes Needed by Phase:**

**Phase 1 – INtime and Test inputs**

* “INtime Product” is currently “INtime Edition”



This will be changed to:

“INtime Product” (IfW DK, IfW Runtime, eVM, dRTOS)

~~“INtime Edition (DK, Runtime)~~

“INtime Version” - 2.5, 6.4, 7, CMP1, CMP3

***effort: 1-3 days, phase: 1, Jesse***

* Test Configuration file - Define core\_intime\_inputs.csv ***effort: .5 day, phase: 1, Team***
  + ***Load parameters – xm/noxm***
  + ***Test parameters***
* Update recipe to accept/pass input parameters
  + -preset1(only tests that apply to this configuration). (May not be needed)
  + Single resource instance reads either specific (intimeapi.csv) and/or generic CSV input (core\_intime\_inputs.csv)

The number of test results in TestRail and the results page will matches piperta execution options + test configuration ***effort: 2 days, phase: 1, Jesse***

* Define “preset configs”
  + default – single node, default memory
  + Preset1 – two nodes, 256mb memory
  + Preset2 – two nodes, 256mb memory, Above 4gb
  + Preset3 – two nodes, 256mb memory, Above 4gb, different core allocation

***effort: .5 day, phase: 1, Team***

* Update result json to include “preset config” setting. ***effort: .5 day, phase: 1***
* “Predefined configuration settings” drop-down on web config page and ability to set via input config. Update installation recipe to configure environment to perform preset configurations (*issue adding node: 13374 -* Create node failure) options ***effort: Unknown (underlying infrastructure not in place for all items specified (core allocation, hw interfaces to nodes)), phase: 1, Jesse***

**Phase 2 – machine inputs**

* A recipe to determine whether the machine is controlled by config process or via web config page. “Fellowship” “One Script to Rule Them All” “Input Config.” If this recipe is selected, no others can be and the form options on the web config page will not be used (Product, version). This recipe will be responsible for reading a machine input and determining which recipes to run. ***effort: 1 week, phase: 2, Jesse***
* Update tests to be executable from a commanding recipe (move recipes -> resources)
  + Clibrary
  + INtimeAPI
  + FIleAPI
  + Gobs
  + Queue
  + Time

***effort: 1 day, phase: 2, Jesse***

**Phase Future – OS/HW inputs**

* The hardware and OS are also specified as inputs and chosen from a pool of resources. Items 1,2 will stay the same for now (a person picks a machine). I propose a rotation of lab machines used each week and a plan for marking what has been tested and when. ***effort: ??, phase: future***
* Tests not otherwise noted above changed to be driven by alternate inputs ***effort: ??, phase: future***
* Improvements to the web config page
  + “INtime Version” and other form fields updated based on fields chosen. e.g., eVM gets 2.5 version only
  + Gray out form fields that do not apply if “input config” selected
  + Prompt to disable fog tasks if update node is clicked and user was changed

***effort: ??, phase: future***