IPMDAR Analyzer

Release 1.1.0

Tecolote Research, Inc.

May 30, 2024

CONTENTS:

CHAPTER

ONE

IPMDAR ANALYZER

1.1 Main Application

1.1.1 Rules Package

1.1.1.1 Contract Performance Dataset (CPD) Test Modules

CPD Level 1 Tests: Basic File Structure Verification

ACWP_ToDate

test_acwp_todate(*acwp_td_df*, *dc_df*, *dm_df*)

Test Description: Validates that the required ACWP_ToDate file meets basic CPD requirements including:

- Confirm that all required fields are present for all entries.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that if DatasetConfiguration's ToDate_TimePhased is true, an entry's ReportingPeriodIDs cannot be null, otherwise, ReportingPeriodIDs must be null.
- Confirm that if DatasetConfiguration's ACWP_ToDate_ByWorkPackage is true, an entry's WorkPackageID cannot be null and ControlAccountID must be null; otherwise, ControlAccountID cannot be null and WorkPackageIDs must be null.
- Confirm that if DatasetConfiguration's Detail_HasDirectValues is true, an entry's Value_Dollars_Direct cannot be null, otherwise, Value_Dollars_Direct must be null.
- Confirm that if DatasetConfiguration's ACWP_ToDate_HasElementOfCostValues is true, an entry's Value_Dollars_LAB, Value_Dollars_MAT, Value_Dollars_ODC, Value_Dollars_SUB cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's ACWP_ToDate_HasElementOfCostValues and Detail_HasDirectValues are true, an entry's Value_Dollars_LAB_Direct, Value_Dollars_MAT_Direct, Value_Dollars_ODC_Direct, Value_Dollars_SUB_Direct cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's Detail_HasIndirectValues is true, an entry's Value_Dollars_OH, Value_Dollars_COM and Value_Dollars_GA cannot be null, otherwise, each must be null.
- Confirm that if not null, an entry's ReportingPeriodID has a value that is less than or equal to DatasetMetadata's ReportingPeriodID.

Parameters

- **acwp_td_df** – A Pandas' dataframe generated from ACWP_ToDate.json
- **dc_df** – A Pandas' dataframe generated from DatasetConfiguration.json
- **dm_df** – A Pandas' dataframe generated from DatasetMetadata.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_acwp_todate_row`(*row*, *dm_period_id*, *file_config*)

Util Method Description: Validates each individual ACWP_ToDate entry and confirms it is valid.

Parameters

- **row** – Pandas dataframe row.
- **file_config** – Dictionary of booleans indicating whether an ACWP_ToDate field is required or not.
- **dm_period_id** – Integer for the ReportingPeriodID from DatasetMetadata.

Returns
Pandas dataframe of alerts indicating issues with the ACWP_ToDate file.

`add_row_error`(*row*, *field*, *message*, *level*, *file_config*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **row** – A Pandas' dataframe row.
- **field** – String indicating the field with the issue
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.
- **file_config** – Dictionary of booleans indicating whether an ACWP_ToDate field is required or not.

Returns
Pandas dataframe with a single alert row.

BCWP_ToDate

`test_bcwp_todate`(*bcwp_td_df*, *dc_df*, *dm_df*)

Test Description: Validates that the required BCWP_ToDate file meets basic CPD requirements including:

- Confirm that all required fields are present for all entries.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that if DatasetConfiguration's ToDate_TimePhased is true, an entry's ReportingPeriodIDs cannot be null, otherwise, ReportingPeriodIDs must be null.
- Confirm that if DatasetConfiguration's BCWP_ToDate_ByWorkPackage is true, an entry's WorkPackageID cannot be null and ControlAccountID must be null; otherwise, ControlAccountID cannot be null and WorkPackageIDs must be null.

- Confirm that if DatasetConfiguration's Detail_HasDirectValues is true, an entry's Value_Dollars_Direct cannot be null, otherwise, Value_Dollars_Direct must be null.
- Confirm that if DatasetConfiguration's BCWP_ToDate_HasElementOfCostValues is true, an entry's Value_Dollars_LAB, Value_Dollars_MAT, Value_Dollars_ODC, Value_Dollars_SUB cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's BCWP_ToDate_HasElementOfCostValues and Detail_HasDirectValues are true, an entry's Value_Dollars_LAB_Direct, Value_Dollars_MAT_Direct, Value_Dollars_ODC_Direct, Value_Dollars_SUB_Direct cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's Detail_HasIndirectValues is true, an entry's Value_Dollars_OH, Value_Dollars_COM and Value_Dollars_GA cannot be null, otherwise, each must be null.
- Confirm that if not null, an entry's ReportingPeriodID has a value that is less than or equal to DatasetMetadata's ReportingPeriodID.

Parameters

- **bcwp_td_df** – A Pandas' dataframe generated from BCWP_ToDate.json
- **dc_df** – A Pandas' dataframe generated from DatasetConfiguration.json
- **dm_df** – A Pandas' dataframe generated from DatasetMetadata.json

Returns

Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_bcwp_todate_row`(*row*, *dm_period_id*, *file_config*)

Util Method Description: Validates each individual BCWP_ToDate entry and confirms it is valid.

Parameters

- **row** – Pandas dataframe row.
- **file_config** – Dictionary of booleans indicating whether an BCWP_ToDate field is required or not.
- **dm_period_id** – Integer for the ReportingPeriodID from DatasetMetadata.

Returns

Pandas dataframe of alerts indicating issues with the BCWP_ToDate file.

`add_row_error`(*row*, *field*, *message*, *level*, *file_config*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **row** – A Pandas' dataframe row.
- **field** – String indicating the field with the issue
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.
- **file_config** – Dictionary of booleans indicating whether an BCWP_ToDate field is required or not.

Returns

Pandas dataframe with a single alert row.

BCWS_ToComplete

`test_bcws_tocomplete`(*bcws_tc_df*, *dc_df*, *dm_df*)

Test Description: Validates that the required BCWP_ToComplete file meets basic CPD requirements including:

- Confirm that all required fields are present for all entries.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that if DatasetConfiguration's BCWS_ToComplete_ByWorkPackage is true, an entry's WorkPackageID cannot be null and ControlAccountID must be null; otherwise, ControlAccountID cannot be null and WorkPackageIDs must be null.
- Confirm that if DatasetConfiguration's Detail_HasDirectValues is true, an entry's Value_Dollars_Direct cannot be null, otherwise, Value_Dollars_Direct must be null.
- Confirm that if DatasetConfiguration's BCWS_ToComplete_HasElementOfCostValues is true, an entry's Value_Dollars_LAB, Value_Dollars_MAT, Value_Dollars_ODC, Value_Dollars_SUB cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's BCWS_ToComplete_HasElementOfCostValues and Detail_HasDirectValues are true, an entry's Value_Dollars_LAB_Direct, Value_Dollars_MAT_Direct, Value_Dollars_ODC_Direct, Value_Dollars_SUB_Direct cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's Detail_HasIndirectValues is true, an entry's Value_Dollars_OH, Value_Dollars_COM and Value_Dollars_GA cannot be null, otherwise, each must be null.
- Confirm that if not null, an entry's ReportingPeriodID has a value that is less than or equal to DatasetMetadata's ReportingPeriodID.

Parameters

- **bcws_tc_df** – A Pandas' dataframe generated from BCWP_ToComplete.json
- **dc_df** – A Pandas' dataframe generated from DatasetConfiguration.json
- **dm_df** – A Pandas' dataframe generated from DatasetMetadata.json

Returns

Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_bcws_tocomp_row`(*row*, *dm_period_id*, *file_config*)

Util Method Description: Validates each individual BCWS_ToComplete entry and confirms it is valid.

Parameters

- **row** – Pandas dataframe row.
- **file_config** – Dictionary of booleans indicating whether an BCWS_ToComplete field is required or not.
- **dm_period_id** – Integer for the ReportingPeriodID from DatasetMetadata.

Returns

Pandas dataframe of alerts indicating issues with the BCWS_ToComplete file.

`create_row_error`(*row*, *field*, *message*, *level*, *file_config*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **row** – A Pandas' dataframe row.
- **field** – String indicating the field with the issue
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.
- **file_config** – Dictionary of booleans indicating whether an BCWP_ToDate field is required or not.

Returns
Pandas dataframe with a single alert row.

BCWS_ToDate

test_bcws_todate(*bcws_td_df*, *dc_df*, *dm_df*)

Test Description: Validates that the required BCWS_ToDate file meets basic CPD requirements including:

- Confirm that all required fields are present for all entries.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that if DatasetConfiguration's ToDate_TimePhased is true, an entry's ReportingPeriodIDs cannot be null, otherwise, ReportingPeriodIDs must be null.
- Confirm that if DatasetConfiguration's BCWS_ToDate_ByWorkPackage is true, an entry's WorkPackageID cannot be null and ControlAccountID must be null; otherwise, ControlAccountID cannot be null and WorkPackageIDs must be null.
- Confirm that if DatasetConfiguration's Detail_HasDirectValues is true, an entry's Value_Dollars_Direct cannot be null, otherwise, Value_Dollars_Direct must be null.
- Confirm that if DatasetConfiguration's BCWS_ToDate_HasElementOfCostValues is true, an entry's Value_Dollars_LAB, Value_Dollars_MAT, Value_Dollars_ODC, Value_Dollars_SUB cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's BCWS_ToDate_HasElementOfCostValues and Detail_HasDirectValues are true, an entry's Value_Dollars_LAB_Direct, Value_Dollars_MAT_Direct, Value_Dollars_ODC_Direct, Value_Dollars_SUB_Direct cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's Detail_HasIndirectValues is true, an entry's Value_Dollars_OH, Value_Dollars_COM and Value_Dollars_GA cannot be null, otherwise, each must be null.
- Confirm that if not null, an entry's ReportingPeriodID has a value that is less than or equal to DatasetMetadata's ReportingPeriodID.

Parameters

- **bcws_td_df** – A Pandas' dataframe generated from BCWS_ToDate.json
- **dc_df** – A Pandas' dataframe generated from DatasetConfiguration.json
- **dm_df** – A Pandas' dataframe generated from DatasetMetadata.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

validate_bcws_todate_row(*row*, *file_config*, *dm_period_id*)

Util Method Description: Validates each individual BCWS_ToDate entry and confirms it is valid.

Parameters

- **row** – Pandas dataframe row.
- **file_config** – Dictionary of booleans indicating whether an BCWS_ToDate field is required or not.
- **dm_period_id** – Integer for the ReportingPeriodID from DatasetMetadata.

Returns
Pandas dataframe of alerts indicating issues with the BCWS_ToDate file.

create_row_error(*row*, *field*, *message*, *level*, *file_config*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **row** – A Pandas' dataframe row.
- **field** – String indicating the field with the issue
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.
- **file_config** – Dictionary of booleans indicating whether an BCWS_ToDate field is required or not.

Returns
Pandas dataframe with a single alert row.

ContractData

test_cpd_contract_data(*df_con_data*)

Test Description: Validates that the required ContractData.json file meets basic CPD requirements including:

- Confirm that all required fields are present for all elements.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Check and report if any dollar values are negative.
- If the effort has been definitized, verify that certain dollar key/values are present.

Parameters
df_con_data – A Pandas' dataframe generated from ContractData.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

contract_data_analytics_check(*df_con_data*)

Analytics Description: Returns specific fields from the ContractData.json file for user's informatiomn:

Parameters
df_con_data – A Pandas' dataframe generated from ContractData.json

Returns
Pandas' dataframe with requested ContractData fields in readable format.

ControlAccounts

`test_cpd_controlAccounts`(*ca_df*)

Test Description: Validates that the required ControlAccounts.json file meets basic CPD requirements including:

- Confirm that all required fields are present for all control accounts.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.

Parameters
ca_df – A Pandas' dataframe generated from ControlAccounts.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_df_row`(*row*, *file_error*)

Util Method Description: Validates each individual control account and confirms it is valid.

Parameters

- **row** – Pandas dataframe row.
- **file_error** – Dictionary structure of a template alert for this specific file.
- **alerts** – Pandas dataframe for all alerts reagrding issues raised during the file test.

Returns
Pandas dataframe of cumulative alerts for the ControlAccounts.json file.

`add_row_error`(*idValue*, *field*, *message*, *level*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **idValue** – String for the ID of the control account.
- **field** – String indicating the field that has the issue
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.
- **alerts** – Pandas dataframe for all alerts regarding issues raised during the file test.

Returns
Pandas dataframe with a single alert row.

DatasetConfigurationAnalyzer

`test_data_config`(*df_config*)

Test Description: Validates that the required DatasetConfiguration.json file meets basic CPD requirements including:

- Confirm that all required fields are present.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.

Parameters
df_config – A Pandas' dataframe generated from DatasetConfiguration.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

EST_ToComplete

`test_est_tocomplete`(*est_tc_df*, *dc_df*, *dm_df*)

Test Description: Validates that the required EST_ToComplete file meets basic CPD requirements including:

- Confirm that all required fields are present for all entries.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that if DatasetConfiguration's EST_ToComplete_ByWorkPackage is true, an entry's WorkPackageID cannot be null and ControlAccountID must be null; otherwise, ControlAccountID cannot be null and WorkPackageIDs must be null.
- Confirm that if DatasetConfiguration's Detail_HasDirectValues is true, an entry's Value_Dollars_Direct cannot be null, otherwise, Value_Dollars_Direct must be null.
- Confirm that if DatasetConfiguration's EST_ToComplete_HasElementOfCostValues is true, an entry's Value_Dollars_LAB, Value_Dollars_MAT, Value_Dollars_ODC, Value_Dollars_SUB cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's EST_ToComplete_HasElementOfCostValues and Detail_HasDirectValues are true, an entry's Value_Dollars_LAB_Direct, Value_Dollars_MAT_Direct, Value_Dollars_ODC_Direct, Value_Dollars_SUB_Direct cannot be null, otherwise, each must be null.
- Confirm that if DatasetConfiguration's Detail_HasIndirectValues is true, an entry's Value_Dollars_OH, Value_Dollars_COM and Value_Dollars_GA cannot be null, otherwise, each must be null.
- Confirm that if not null, an entry's ReportingPeriodID has a value that is less than or equal to DatasetMetadata's ReportingPeriodID.

Parameters

- **est_tc_df** – A Pandas' dataframe generated from EST_ToComplete.json
- **dc_df** – A Pandas' dataframe generated from DatasetConfiguration.json
- **dm_df** – A Pandas' dataframe generated from DatasetMetadata.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_est_tocomp_row`(*row*, *dm_period_id*, *file_config*)

Util Method Description: Validates each individual EST_ToComplete entry and confirms it is valid.

Parameters

- **row** – Pandas dataframe row.
- **file_config** – Dictionary of booleans indicating whether an EST_ToComplete field is required or not.
- **dm_period_id** – Integer for the ReportingPeriodID from DatasetMetadata.

Returns
Pandas dataframe of alerts indicating issues with the EST_ToComplete file.

`add_row_error`(*row*, *field*, *message*, *level*, *file_config*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **row** – A Pandas' dataframe row.
- **field** – String indicating the field with the issue.
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.
- **file_config** – Dictionary of booleans indicating whether an BCWP_ToDate field is required or not.

Returns
Pandas dataframe with a single alert row.

Metadata

`test_cpd_metadata`(*metadata_df*)

Test Description: Validates that the required DatasetMetadata.json file meets basic CPD requirements including:

- Confirm that all required fields are present for all control accounts.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that ContractorIDCode is null unless ContractorIDCodeTypeID is not null.
- Confirm that EVMSAcceptanceDate is null unless EVMSAccepted is true.
- Check and report if certain general Contract/Effort information fields are missing or empty
- Check if SecurityMarking if set to 'unclassified'.

Parameters
metadata_df – A Pandas' dataframe generated from DatasetMetadata.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`check_fields`(*metadata_df*)

Util Method Description: Performs checks on the DatasetMetadata dataframe to ensure all required colulment are present and log the existence of any unapproved fields.

Parameters
`metadata_df` – A Pandas' dataframe generated from DatasetMetadata.json

Returns
Two lists, one containing the name's of fields that are required to be in DatasetMetadata.json but are not, and one conatining the name's of fields present in the dataframe that should not be there.

`dataset_metadata_analytics_check`(*metadata_df*)

Analytics Description: Returns specific fields from the DatasetMetadata.json in a specific order file for user's informatiomn:

Parameters
`metadata_df` – A Pandas' dataframe generated from DatasetMetadata.json

Returns
Pandas' dataframe with requested DatasetMetadata fields in readable format.

OBS

`test_cpd_obs`(*obs_df*)

Test Description: Validates that the required OBS.json file meets basic CPD requirements including:

- Confirm that all required fields are present for all elements.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm the first element on record has a Level of 1.
- Confirm that the first element on record has no ParentID and that all subsequent elements have a ParentID

Parameters
`obs_df` – A Pandas' dataframe generated from OBS.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`return_obs_row_ID`(*row*)

Util Method Description: Isolates and returns an OBS element's ID field if it exists or 'N/A' if it does not.

Parameters
`row` – Pandas' dataframe row.

Returns
String values of either the OBS element's ID or N/A.

`validate_level_type`(*row*, *structure_col*)

Util Method Description: Validates the Level of each OBS element in the file.

Parameters

- `row` – Pandas' dataframe row.
- `structure_col` – Pandas' dataframe row.

Returns
Boolean indicating whether an OBS element's Level field is valid.

SummaryPerformance

`test_cpd_summaryperformance`(*sp_df*)

Test Description: Validates that the required SummaryPerformance.json file meets basic CPD requirements including:

- Confirm that all required fields are present for all elements.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that the SummaryPerformance.json file contains one entry with a SummaryElementID equal to 'PMB'.
- Verify that the 'PMB' line contain a valid dollar value for fields BAC_Dollars and EAC_Dollars.

Parameters
sp_df – A Pandas' dataframe generated from SummaryPerformance.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`check_columns`(*sp_df*)

Util Method Description: Performs checks on the SummaryPerformance dataframe to ensure all required colulment are present and log the existence of any unapproved fields.

Parameters
sp_df – A Pandas' dataframe generated from SummaryPerformance.json

Returns
Two lists, one containing the name's of fields that are required to be in SummaryPerformance.json but are not, and one conatining the name's of fields present in the dataframe that should not be there.

`check_rows`(*sp_df*, *file_error*)

Util Method Description: Performs checks on an individual Summary element line, validating the existence and data types for the included fields.

Parameters

- **sp_df** – A Pandas' dataframe generated from SummaryPerformance.json
- **file_error** – A template error alert Dictionary that is used to generate alert messages.

Returns
Pandas dataframe of alerts indicating issues with an individual Summary Element line.

`check_PMB`(*PMB_row*, *file_error*)

Util Method Description: Performs a set of checks unique to the SummaryPerformance's PMB line.

Parameters

- **PMB_row** – Pandas dataframe row, specifically the PMB line
- **file_error** – A template error alert Dictionary that is used to generate alert messages.

Returns
Pandas dataframe of alerts indicating issues with an individual Summary Element line.

`check_field_decimal`(*field*, *fieldName*, *file_error*)

Util Method Description: Performs checks and validates data type specifically for fields that can be a decimal or an integer.

Parameters

- **field** – Integer or float for the field value to be tested.
- **fieldName** – String for the name of the field to be tested.
- **file_error** – A template error alert Dictionary that is used to generate alert messages.

Returns
Pandas dataframe of alerts indicating issues with an individual Summary Element line.

`summary_perf_analytics_check`(*sp_df*)

Analytics Description: Returns specific fields from the SummaryPerformance.json in a specific order file for user's informatiomn:

Parameters
sp_df – A Pandas' dataframe generated from SummaryPerformance.json

Returns
Pandas' dataframe with requested SummaryPerformance fields in readable format.

SummaryIndirectPerformance_ToDate

`test_cpd_summaryIndirectPerformance_td`(*cpd*)

Test Description: Validates that the optional SummaryIndirectPerformance_ToDate.json file meets basic CPD requirements including:

- Verify if the SummaryIndirectPerformance_ToDate.json file is part of the CPD and that it is not empty.
- Confirm that all required fields are present for all summary indirect elements.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that each summary indirect elements has a ReportingPeriodID less than or equal to the ReportingPeriodID located in the DatasetMetadata.json file.

Parameters
cpd – Dictionary of Panda's dataframes for the entire CPD.

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_summ_perf_td_row`(*row*, *file_error*, *periodId*, *time_phased*)

Util Method Description: Validates each individual entry in the SummaryIndirectPerformance_ToDate file and confirms it is valid.

Parameters
row – Pandas dataframe row representing a single work package.

SummaryIndirectPerformance_ToComplete

`test_cpd_summaryIndirectPerformance_tc`(*cpd*)

Test Description: Validates that the optional SummaryIndirectPerformance_ToComplete.json file meets basic CPD requirements including:

- Verify if the SummaryIndirectPerformance_ToComplete.json file is part of the CPD and that it is not empty.
- Confirm that all required fields are present for all summary indirect elements.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that each summary indirect elements has a ReportingPeriodID greater than the ReportingPeriodID located in the DatasetMetadata.json file.

Parameters
cpd – Dictionary of Panda's dataframes for the entire CPD.

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_summ_perf_tc_row`(*row*, *file_error*, *periodId*)

Util Method Description: Validates each individual entry in the SummaryIndirectPerformance_ToComplete file and confirms it is valid.

Parameters
row – Pandas dataframe row representing a single work package.

WBS

`test_cpd_wbs`(*wbs_df*)

Test Description: Validates that the required WBS.json file meets basic CPD requirements including:

- Confirm that all required fields are present for all elements.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm the first element on record has a Level of 1.
- Confirm that the first element on record has no ParentID and that all subsequent elements have a ParentID

Parameters
wbs_df – A Pandas' dataframe generated from WBS.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`return_wbs_row_ID`(*row*)

Util Method Description: Isolates and returns an WBS element's ID field if it exists or 'N/A' if it does not.

Parameters
row – Pandas' dataframe row.

Returns
String values of either the OBS element's ID or N/A.

`validate_level_type`(*row*, *structure_col*)

Util Method Description: Validates the Level of each WBS element in the file.

Parameters

- **row** – Pandas' dataframe row.
- **structure_col** – Pandas' dataframe row.

Returns
Boolean indicating whether an OBS element's Level field is valid.

WorkPackages

`test_cpd_work_packages`(*cpd*)

Test Description: Validates that the optional WorkPackages.json file meets basic CPD requirements including:

- Verify that the WorkPackages.json file is part of the CPD and that it is not empty.
- Confirm that all required fields are present for all control accounts.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that the value OtherEarnedValueTechnique for an individual work package must be null unless EarnedValueTechniqueID has a value of 'OTHER_DISCRETE' or 'FIXED_X_Y'.

Parameters
cpd – Dictionary of Panda's dataframes for the entire CPD.

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_wp_row`(*row*, *file_error*)

Util Method Description: Validates each individual work package and confirms it is valid.

Parameters
row – Pandas dataframe row representing a single work package.

`add_row_error`(*idValue*, *field*, *message*, *level*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **idValue** – String for the ID of the control account.
- **field** – String indicating the field that has the issue
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.

ReprogrammingAdjustments

`test_cpd_ReprogrammingAdjustments`(*cpd*)

Test Description: validates that the optional ReporgrammingAdjustments.json file meets the basic requirements including:

- Verify if the ReporgrammingAdjustments.json file is part of the CPD and that is not empty.
- Verify that all the required fields are present.
- Verify that no non-approved keys are present in each entry.
- Verify that all provided values are of the specific type.

Parameters
cpd – Dictionary of Panda's dataframes for the entire CPD.

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_reprogramming_adjustments_row`(*row*, *filer_error*)

UTIL Method Description: Validates each individual entry in the ReprogrammingAdjustments file and confirms it is valid.

Parameters
row – Pandas dataframe row.

CPD Level 2 Tests: WBS & OBS Structures

Hierarchical Element Structure Evaluation

`generate_element_structure_df`(*element_df*, *type*)

Util Method Description: Builds an element structure dataframe from either a WBS or OBS dataframe.

Parameters

- **element_df** – Pandas dataframe of either WBS or OBS file
- **type** – string value representing file type, i.e. "obs" or "wbs".

Returns
Pandas dataframe containing information necessary to perform later structural validations

`verify_element_structure`(*structure_df*)

Test Method Description: Verifies that each element's declared ParentID matches the one derived from the structure.

Parameters
structure_df – Pandas dataframe of WBS or OBS structure generated by generate_element_structure_df function

Returns
Dictionary of string to Pandas dataframe key/value pairs that correspond to the alerts and messages generated by the test

`verify_node_levels`(*structure_df*, *type*)

Test Method Description: Verifies the changes in level between parent-child pairs derived from the structure.

Parameters

- **structure_df** – Pandas dataframe of WBS or OBS structure generated by generate_element_structure_df function
- **type** – string value representing file type, i.e. "obs" or "wbs".

Returns

Dictionary of string to Pandas dataframe key/value pairs that correspond to the alerts and messages generated by the test

`verify_control_accounts`(*structure_df*, *control_accounts_df*, *type*)

Test Method Description: Verifies the mapping between element pairs in WBS/OBS dataframe and control accounts dataframe.

Parameters

- **structure_df** – Pandas dataframe of WBS or OBS structure generated by generate_element_structure_df function
- **control_accounts_df** – Pandas dataframe of ControlAccounts file
- **type** – string value representing file type, i.e. "obs" or "wbs".

Returns

Dictionary of string to Pandas dataframe key/value pairs that correspond to the alerts and messages generated by the test

CPD Level 3 Tests: Cross-File Data Verification and Analysis

ContractData Order of Magnitude Tests

`test_contract_order_mag`(*df_con_om*)

Test Description: Validates the order of magnitude and ManagementEAC order for ContractData values by checking that all ContractData numbers are within a 20% order of magnitude of each other and that ManagementEAC_MostLikely is between BestCase and WorstCase.

Parameters

df_con_om – Pandas dataframe generated from ContractData.json.

Returns

Pandas dataframe of file alerts.

ContractData Date Field Verification

`verify_contractData_dates`(*ca_df*)

Test Description: Verifies date inputs in ContractData, including Start and Completion Dates comparisons, in particular:

- Verifies all date values via the check_date_valid function.
- Compares and validates the Start Date with the Completion Date via the compare_two_dates function.
- Compares and validates the Definition Date with the Completion Date via the compare_two_dates function.

- Compares and validates the Baseline Date with the other Completion Dates via the compare_two_dates function.

Parameters
ca_df – Pandas dataframe generated from ContractData.json file.

Returns
Pandas dataframe of file alerts.

`compare_two_dates`(*start_date*, *start_name*, *compl_date*, *compl_name*)

Util Method Description: Compares and validates the difference between two dates.

Parameters

- **start_date** – String value corresponding to the inputted start date.
- **start_name** – String value corresponding to the field that the inputted start date was pulled from.
- **compl_date** – String value corresponding to the inputted end date.
- **compl_name** – String value corresponding to the field that the inputted end date was pulled from.

Returns
Pandas dataframe of file alerts.

ControlAccounts Date Range Analytics

`find_cpd_date_range`(*ca_df*)

Analytics Description: Determines the date range of a contract via the CPD's ControlAccounts file and logs important analytics, including:

- The earliest BaselineStartDate.
- The latest BaselineStartDate.

Parameters
ca_df – Pandas dataframe generated from ControlAccounts.json.

Returns
Pandas dataframe of file analytics.

ControlAccounts-to-ContractData Date Field Comparisons

`verify_controlAccounts_contractData_dates`(*ca_df*, *cd_df*)

Test Description: Validates Completion Dates between ControlAccounts and ContractData files by ensuring that all BaselineEndDates in ControlAccounts are less then or equal to the BaselineCompletionDate in ContractData

Parameters

- **ca_df** – Pandas dataframe generated from ControlAccounts.json.
- **dc_df** – Pandas dataframe generated from ContractData.json.

Returns
Pandas dataframe of file alerts.

`verify_ca_cd_baselineCompletionDate`(*ca_df*, *cd_df*)

Test Description: Verifies that the latest BaselineEndDate in ControlAccounts is equal to the BaselineCompletionDate in ContractData.

Parameters

- **ca_df** – Pandas dataframe generated from ControlAccounts.json.
- **dc_df** – Pandas dataframe generated from ContractData.json.

Returns
Pandas dataframe of file alerts.

DatasetConfiguration Element Consistency Checks

`data_config_boolean_check`(*config_info*, *return_errors*)

Test Description: Validates consistency of WorkPackages and ElementsOfCosts usage in DatasetConfiguration or reports WorkPackage and ElementOfCost usage in DatasetConfiguration.

- If return_errors is set to True, check consistency of WorkPackages and ElementsOfCosts usage in DatasetConfiguration.
- If return_errors is set to False, report WorkPackage and ElementsOfCost Usage in DatasetConfiguration.

Parameters

- **config_info** – Pandas dataframe generated from DatasetConfiguration.json.
- **return_errors** – Boolean value indicating what functionality to perform.

Returns
Pandas dataframe of file alerts if return_errors is set to true, otherwise returns Pandas dataframe of file analytics.

DatasetConfiguration Data Analytics

`data_config_analytics_check`(*config_info*)

Analytics Description: Parses config_info dataframe and generates analytics, including:

- Which values are reporting by Work Packages.
- Which values are reporting by Control Accounts.
- Which values are reporting Element of Cost.
- The boolean value of 'ToDate_TimePhased'.

Parameters
config_info – Pandas dataframe generated from DatasetConfiguration.json.

Returns
Pandas dataframe of file analytics.

EV Metrics Files Value Dollar Total Checks

`test_bcws_todate_value_dollars`(*df_bcws*, *df_dc*)

Test Description: Validates Value_Dollars calculations included in the check_row_value_dollars function for BCWS_ToDate file.
The calculations are the following:

- Determines that the sum of all Total_Value_Direct numbers equals Value_Dollars for each row.
- Determines the Value_Dollars calculated ratios by checking that the DatasetConfiguration files NonAdd_OH, NonAdd_COM, NonAdd_GA are present and also that the values of the Value_Dollars groupings, LAB, MAT, SUB, and ODC are both present and not zero.
- Determines that the calculated ratios are not below 1 or above 2.5.

Parameters

- **df_bcws** – Pandas dataframe generated from BCWS_ToDate.json.
- **df_dc** – Pandas dataframe generated from DatasetConfiguration.json.

Returns
Pandas dataframe of file alerts.

`test_bcwp_todate_value_dollars`(*df_bcwp*, *df_dc*)

Test Description: Validates Value_Dollars calculations included in the check_row_value_dollars function for BCWP_ToDate file.
The calculations are the following:

- Determines that the sum of all Total_Value_Direct numbers equals Value_Dollars for each row.
- Determines the Value_Dollars calculated ratios by checking that the DatasetConfiguration files NonAdd_OH, NonAdd_COM, NonAdd_GA are present and also that the values of the Value_Dollars groupings, LAB, MAT, SUB, and ODC are both present and not zero.
- Determines that the calculated ratios are not below 1 or above 2.5.

Parameters

- **df_bcwp** – Pandas dataframe generated from BCWP_ToDate.json.
- **df_dc** – Pandas dataframe generated from DatasetConfiguration.json.

Returns
Pandas dataframe of file alerts.

`test_acwp_todate_value_dollars`(*df_acwp*, *df_dc*)

Test Description: Validates Value_Dollars calculations for ACWP_ToDate file by checking that the sum of all Total_Value_Direct numbers equals Value_Dollars for each row, per the check_row_value_dollars function.

Parameters

- **df_acwp** – Pandas dataframe generated from ACWP_ToDate.json.
- **df_dc** – Pandas dataframe generated from DatasetConfiguration.json.

Returns
Pandas dataframe of file alerts.

`test_bcws_tocomplete_value_dollars`(*df_bcws_tc*, *df_dc*)

Test Description: Validates Value_Dollars calculations included in the check_row_value_dollars function for BCWS_ToComplete file.
The calculations are the following:

- Determines that the sum of all Total_Value_Direct numbers equals Value_Dollars for each row.
- Determines the Value_Dollars calculated ratios by checking that the DatasetConfiguration files Non-Add_OH, NonAdd_COM, NonAdd_GA are present and also that the values of the Value_Dollars groupings, LAB, MAT, SUB, and ODC are both present and not zero.
- Determines that the calculated ratios are not below 1 or above 2.5.

Parameters

- **df_bcws_tc** – Pandas dataframe generated from BCWS_ToComplete.json.
- **df_dc** – Pandas dataframe generated from DatasetConfiguration.json.

Returns
Pandas dataframe of file alerts.

`test_est_tocomplete_value_dollars`(*df_est_tc*, *df_dc*)

Test Description: Validates Value_Dollars calculations included in the check_row_value_dollars function for EST_ToComplete file.
The calculations are the following:

- Determines that the sum of all Total_Value_Direct numbers equals Value_Dollars for each row.
- Determines the Value_Dollars calculated ratioa by checking that the DatasetConfiguration files Non-Add_OH, NonAdd_COM, NonAdd_GA are present and also that the values of the Value_Dollars groupings, LAB, MAT, SUB, and ODC are both present and not zero.
- Determines that the calculated ratioa are not below 1 or above 2.5.

Parameters

- **df_est_tc** – Pandas dataframe generated from EST_ToComplete.json.
- **df_dc** – Pandas dataframe generated from DatasetConfiguration.json.

Returns
Pandas dataframe of file alerts.

`check_row_value_dollars`(*row*, *file*, *has_eoc*, *has_direct_values*, *has_nonadd_oh*, *has_nonadd_com*, *has_nonadd_ga*)

Util Method Description: Checks that the sum of all Total_Value_Direct numbers equals Value_Dollars.
Additionaly, this method also determines the ratios for the Value_Dollars grouping LAB, MAT, SUB, and ODC.

Parameters

- **row** – A row of a Pandas dataframe generated by pandas.DataFrame.itertuples.
- **file** – String value representing the name of a file.
- **has_eoc** – Boolean value corresponding to whether or not the file has element of cost values.

- **has_direct_values** – Boolean value corresponding to whether or not the file has direct values.
- **has_nonadd_oh** – String value representing the name of a file.
- **has_nonadd_com** – String value representing the name of a file.
- **has_nonadd_ga** – String value representing the name of a file.

Returns
Dictionary corresponding to any value errors generated.

SummaryPerformance-to-ContractData TAB and CBB Validation

verify_summaryPerfromance_contractData_budgets(*sp_df*, *cd_df*)

Test Description: Validates TAB and CBB between SummaryPerformance and ContractData by checking that Contract Budget Base (CBB) values from Summary Performance match with the manually entered Total Allocated Budget (TAB) and CBB values from ContractData.

Parameters

- **sp_df** – Pandas dataframe generated from SummaryPerformance.json file.
- **cd_df** – Pandas dataframe generated from ContractData.json file.

Returns
Pandas dataframe of file alerts.

SummaryPerformance-to-ContractData PMB Line Order of Magnitude Checks

verify_summaryPerfromance_contractData_costs(*sp_df*, *cd_df*)

Test Description: Validates that PMB Line EAC and BAC_Dollars are in the same 20% order or magnitude as ContractData's values, and also that:

- The EAC_Dollars value for the PMB line in SummaryPerformance equals ContractData's ManagementEAC_MostLikely.
- The BAC_Dollars value for the PMB line in SummaryPerformance is inside the order of magnitude for values in ContractData.
- The EAC_Dollars value for the PMB line in SummaryPerformance is inside the order of magnitude for values in ContractData.
- Verify that the BAC_DOllars value for the PMB line in SummaryPerformance is greater than or equal to Target_Price in ContractData

Parameters

- **sp_df** – Pandas dataframe generated from SummaryPerformance.json file.
- **cd_df** – Pandas dataframe generated from ContractData.json file.

Returns
Pandas dataframe of file alerts.

WorkPackages Date Field Analytics

`evaluate_work_packages_dates`(*wp_df*)

Analytics Description: Determines the date range of a contract via the CPD's ControlAccounts, logging analytics such as:

- The earliest BaselineStartDate.
- The latest BaselineStartDate.

Parameters
wp_df – Pandas dataframe generated from WorkPackages.json file.

Returns
Pandas dataframe of file analytics.

WorkPackages-to-ContractData End Date Verifications

`check_workpackage_contractdata_date`(*wp_df*, *cd_df*)

Test Description: Verifies that WorkPackages' BaselineEndDates are before ContractData's BaselineCompletionDate.

Parameters

- **wp_df** – Pandas dataframe generated from WorkPackages.json file.
- **cd_df** – Pandas dataframe generated from ContractData.json file.

Returns
Pandas dataframe of file alerts.

`compare_latest_wp_baselnenddt_to_cd_baselncompdt`(*wp_df*, *cd_df*)

Test Description: Verifies that the latest WorkPackages' BaselineEndDate is equal to ContractData's BaselineCompletionDate.

Parameters

- **wp_df** – Pandas dataframe generated from WorkPackages.json file.
- **cd_df** – Pandas dataframe generated from ContractData.json file.

Returns
Pandas dataframe of file alerts.

CPD Level 4: Summary Elements Data Verification

SummaryPerformance Element Presence Tests

`check_summary_performance_elements`(*sp_df*, *dc_df*)

Test Description: Evaluates the SummaryPerformance file and reports the existence of each Summary Element other than the required PMB line, in particular:

- Check for the existence of a 'OH' Element. If missing report either a Medium or Low alert based on the status of the OH No Add flag.
- Check for the existence of a 'COM' Element. If missing report either a Medium or Low alert based on the status of the COM No Add flag.

- Check for the existence of a 'GA' Element. If missing report either a Medium or Low alert based on the status of the GA No Add flag.
- Check for the existence of a 'UB' Element. If missing report a Low Alert.
- Check for the existence of a 'MR' Element. If missing report a Low Alert.

Parameters

- **sp_df** – Pandas dataframe generated from SummaryPerformance.json file.
- **dc_df** – Pandas dataframe generated from DatasetConfiguration.json file.

Returns

Pandas dataframe of file alerts.

SummaryPerformance Element Required Content Checks

`verify_sp_element_contents`(*sp_df*, *dc_df*)

Test Description: Evaluates the SummaryPerformance file bases on the Detail_HasIndirectValues flag in DatasetConfiguration. Following tests are only performed if Detail_HasIndirectValues is True:

- If True, OH, COM, GA Elements MUST be present in the SummaryPerformance.json file, report High Alert if they are not.
- OH, COM, GA MUST have values: BAC_Dollars, EAC_Dollars, BCWS_CumulativeToDate_Dollars, BCWP_CumulativeToDate_Dollars, ACWP_CumulativeToDate_Dollars, report High Alert if missing.
- The above values should not be negative or 0, report Medium Alert if they are.

Parameters

- **sp_df** – Pandas dataframe generated from SummaryPerformance.json file.
- **dc_df** – Pandas dataframe generated from DatasetConfiguration.json file.

Returns

Pandas dataframe of file alerts.

1.1.1.2 Schedule Performance Dataset (SPD) Test Module

SPD Level 1 Tests: Basic File Structure Verification

DatasetMetadata

`test_spd_metadata`(*metadata_df*)

Test Description: Validates that the required DatasetMetadata.json file meets basic SPD requirements including:

- Confirm that all required fields are present for all control accounts.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that ContractorIDCode is null unless ContractorIDCodeTypeID is not null.
- Confirm that EVMSAcceptanceDate is null unless EVMSAccepted is true.
- Check if SecurityMarking if set to 'unclassified'.

Parameters
metadata_df – A Pandas' dataframe generated from DatasetMetadata.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`check_fields`(*metadata_df*)

Util Method Description: Performs checks on the DatasetMetadata dataframe to ensure all required colulment are present and log the existence of any unapproved fields.

Parameters
metadata_df – A Pandas' dataframe generated from DatasetMetadata.json

Returns
Two lists, one containing the name's of fields that are required to be in DatasetMetadata.json but are not, and one conatining the name's of fields present in the dataframe that should not be there.

`dataset_metadata_analytics_check`(*metadata_df*)

Analytics Description: Returns specific fields from the DatasetMetadata.json in a specific order file for user's information:

Parameters
metadata_df – A Pandas' dataframe generated from DatasetMetadata.json

Returns
Pandas' dataframe with requested DatasetMetadata fields in readable format.

ProjectScheduleData

`test_project_schedule_data`(*psd_df*)

Test Description: Validates that the required ProjectScheduleData.json file meets basic SPD requirements including:

- Confirm that all required fields are present.
- Confirm that all field values are of the correct type.
- Check and report the presence of any non-approved fields.

Parameters
psd_df – A Pandas dataframe generated from ProjectScheduleData.json.

Returns
Pandas dataframe of alerts indicating issues with the ProjectScheduleData file.

`project_schedule_data_analytics_check`(*psd_df*)

Analytics Description: Returns specific fields from the ProjectScheduleData.json file for user's information:

Parameters
psd_df – A Pandas' dataframe generated from ProjectScheduleData.json

Returns
Pandas' dataframe with requested ProjectScheduleData fields in readable format.

TaskScheduleData

`test_task_schedule_data_file`(*tsd_df*)

Test Description: Validates that the required TaskScheduleData.json file meets basic SPD requirements including:

- Confirm that all required fields are present for all control accounts.
- Confirm that all field' values are of the correct type.
- Check and report the presence of any non-approved fields.
- Confirm that each Tasks baseline data entries are either all null or all reported.

Parameters
tsd_df – A Pandas' dataframe generated from TaskScheduleData.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`validate_tsd_row`(*row*)

`add_row_error`(*row*, *field*, *message*, *level*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **row** – A Pandas' dataframe row.
- **field** – String indicating the field with the issue
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.

Returns
Pandas dataframe with a single alert row.

Tasks

`test_spd_tasks`(*tasks_df*)

Test Description: Validates that the required Tasks.json file meets basic SPD requirements including:

- Confirm that all required fields are present for all tasks.
- Confirm that all field values are of the correct type.
- Confirm any field values with key constraints match with their respective enumeration values.
- Check and report the presence of any non-approved fields.
- Confirm that TaskPlanningLevelID is not null if TaskTypeID has a value of ACTIVITY.
- Confirm that OtherEarnedValueTechnique is null unless EarnedValueTechniqueID has a value of OTHER_DISCRETE or FIXED_X_Y.

Parameters
tasks_df – A Pandas' dataframe generated from Tasks.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`check_tasks_row`(*row*)

Util Method Description: Validates each individual task and confirms it is valid.

Parameters
row – Pandas dataframe row representing a single task.

Returns
An array of alerts from the given row.

`add_row_error`(*idValue*, *field*, *message*, *level*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **idValue** – String for the ID of the control account.
- **field** – String indicating the field that has the issue
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.

TaskOutlineStructure

`test_spd_tos`(*tos_df*)

Test Description: Validates that the required TaskOutlineStructure.json file meets basic SPD requirements including:

- Confirm that all required fields are present for all tasks.
- Confirm that all field values are of the correct type.
- Confirm any field values with key constraints match with their respective enumeration values.
- Check and report the presence of any non-approved fields.
- Confirm that the file's first entry has a Level of 1
- Confirm that file entries with a Level of 1 do not have a ParentTaskID and that all other entries do.
- Verify that all file entries with a Parent Task ID have a Parent Task that is at a higher level than they are.

Parameters
tos_df – A Pandas' dataframe generated from TaskOutlineStructure.json

Returns
Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`check_tos_row`(*row*)

Util Method Description: Validates each individual task and confirms it is valid.

Parameters
row – Pandas dataframe row representing a single task.

Returns
An array of alerts from the given row.

`verify_file_structure`(*tos_df*)

Util Method Description: Validates the overall hierachical structure of the Task Outline Structure file.

Parameters
`tos_df` – Pandas dataframe representing the TaskOutlineStructure.json file.

Returns
An array of alerts from the given row.

`add_row_error`(*idValue*, *field*, *message*, *level*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- `idValue` – String for the ID of the control account.
- `field` – String indicating the field that has the issue
- `message` – String for the alert message's text.
- `level` – String for the level indicating the severity of the alert.

SPD Level 2 Tests: Project Schedule and Task Validation

ProjectScheduleData Date Field Timeline Checks

`test_project_schedule_date`(*psd_df*)

Test Description: Verifies that ProjectScheduleData's start dates are before finish dates including Actual Start/Finish Dates being before or equal to the StatusDate.

Parameters
`psd_df` – Pandas dataframe generated from ProjectScheduleData.json file.

Returns
Pandas dataframe of file alerts.

TaskScheduleData Data Analytics

`task_schedule_data_analytics`(*tsd_df*)

Analytics Description: Analyzes the TaskScheduleData.json file and returns a count of all Tasks on the Critical Path.

Parameters
`tsd_df` – Pandas dataframe generated from TaskScheduleData.json.

Returns
Pandas dataframe of file analytics.

Tasks Activity and Milestone 60-Day Start Date Checks

`validate_actual_start_dates`(*psd_df*, *tasks_df*, *tsd_df*)

Test Description: Validates that for Tasks with a TaskTypeID of ACTIVITY or MILESTONE, Actual Start Date is both given, and within 60 days prior to StatusDate, only if the Current Start Date is also within 60 days prior to StatusDate.

Parameters

- **psd_df** – Pandas dataframe generated from ProjectScheduleData.json.
- **tasks_df** – Pandas dataframe generated from Tasks.json.
- **tsd_df** – Pandas dataframe generated from TaskScheduleData.json.

Returns

Dictionary of string to Pandas dataframe key/value pairs that correspond to the alerts and messages generated by the test.

Tasks Data Analytics

`tasks_analytics`(*tasks_df*)

Analytics Description: Analyzes the Tasks.json file and returns the total number of all tasks. Also provides the total of the tasks broken down by each TaskTypeID: Activity, Milestone, Summary, and Hammock.

Parameters

tsd_df – Pandas dataframe generated from Tasks.json.

Returns

Pandas dataframe of file analytics.

Tasks-to-TaskScheduleData Activity and Milestone Date Tests

`tasks_tsd_date_test`(*tasks_df*, *tsd_df*)

Test Description: Test joins the Tasks and TaskScheduleData dataframes by the Task ID filtered by Task Types: ACTIVITY and MILESTONE. Then it tests on any listed Duration and Dates depending on its Task Type.

- **For all Activities:**
  - Both CurrentDuration and BaselineDuration should NOT be zero.
  - CurrentStartDate and CurrentFinishDate should not equal each other if CurrentDuration is greater than 1.
  - BaselineStartDate and BaselineFinishDate should not equal each other if BaselineDuration is greater than 1.
- **For all Milestones:**
  - Both CurrentDuration and BaselineDuration should be zero.
  - CurrentStartDate and CurrentFinishDate should equal each other.
  - BaselineStartDate and BaselineFinishDate should equal each other.

Parameters

- **tasks_df** – Pandas dataframe generated from Tasks.json.

- **tsd_df** – Pandas dataframe generated from TaskScheduleData.json.

Returns
Pandas dataframe of file alerts.

validate_date_duration(*tsd_df*)

Util Method Description[DISABLED]: Validates start and finish date pairs, specifically that:

- Required fields CurrentStartDate and CurrentFinishDate have a difference equal to the value of CurrentDuration with a one day buffer.
- Nullable fields BaselineStartDate and BaselineFinishDate, if present, have a difference equal to the value of BaselineDuration with a one day buffer.

Parameters
tsd_df – Pandas dataframe generated from TaskScheduleData.json.

Returns
Pandas dataframe of file alerts.

validate_start_finish_date_pairs(*tsd_df*)

Util Method Description Verifies that the finish date does not come before the start date for each of the following start/finish date pairs:

- CurrentStartDate and CurrentFinishDate,
- EarlyStartDate and EarlyFinishDate,
- LateStartDate and LateFinishDate,
- BaselineStartDate and BaselineFinishDate,
- ActualStartDate and ActualFinishDate.

Parameters
tsd_df – Pandas dataframe generated from TaskScheduleData.json.

Returns
Pandas dataframe of file alerts.

format_day_plurality(*difference*)

Util Method Description Generates and returns the value of difference followed by 'day' if the value of difference is equal to 1 and 'days' if the value of difference is greater than 1.

Parameters
difference – A positive non-zero integer.

Returns
A string formatted per the method description.

add_row_error(*idValue*, *field*, *message*, *level*)

Util Method Description: Generates and returns a row error dataframe.

Parameters

- **idValue** – String for the ID of the control account.
- **field** – String indicating the field that has the issue
- **message** – String for the alert message's text.
- **level** – String for the level indicating the severity of the alert.

TaskOutLineStructre-to-Tasks Summary Alignment Checks

`test_spd_tos_task_alignment`(*tos_df*, *tasks_df*)

Test Description: Verifies that each entry's TaskID in the TaskOutlineStructure.json file has a corresponding Task in the Tasks.json file and that that Task is of Type 'SUMMARY':

Parameters

- **tos_df** – A Pandas' dataframe generated from TaskOutlineStructure.json
- **tasks_df** – A Pandas' dataframe generated from Tasks.json

Returns

Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

TaskScheduleData-to-ProjectScheduleData Date Range Checks

`verify_tasks_dates`(*tsd_df*, *psd_df*)

Test Description: Perform date match checks for corresponding values in the Project and TaskScheduleData files including:

- Confirm that the earliest CurrentStartDate in TaskScheduleData matches the CurrentStartDate listed in ProjectScheduleData.
- Confirm that the latest CurrentFinishDate in TaskScheduleData matches the CurrentFinishDate listed in ProjectScheduleData if listed.
- Confirm that the earliest BaselineStartDate in TaskScheduleData matches the BaselineStartDate listed in ProjectScheduleData if listed.
- Confirm that the latest BaselineFinishDate in TaskScheduleData matches the BaselineFinishDate listed in ProjectScheduleData if listed.
- Confirm that the earliest ActualStartDate in TaskScheduleData matches the ActualStartDate listed in ProjectScheduleData if listed.
- Confirm that all Task Actual Start/Finish dates if listed are on or before the Project's Status Date.

Parameters

- **tsd_df** – Panda's dataframe generated from TaskScheduleData.json.
- **psd_df** – Panda's dataframe generated from ProjectScheduleData.json.

Returns

Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`add_alert_message`(*row*, *psd_date*, *psd_field*, *dateType*, *futureAlert=False*)

Util Method Description: Generates and returns a row alert dataframe.

Parameters

- **row** – A Pandas' dataframe row.
- **psd_date** – The project level date that the Task date does not match.
- **psd_field** – String for the row column with the non-matching Task date.

- **dateType** – String indicating whether this is either the earliest or latest date in TaskScheduleData.
- **futureAlert** – Optional boolean to indicate this alert is for a specific row with an actual date after the ProjectStatusDate.

Returns
dictionary representing a single alert row.

1.1.1.3 Cost Schedule Integration (CSI) Test Module

CSI Level 1 Tests: Cost/Schedule Integration Analytics

CPD/SPD DatasetMetadata Joined Analytics

csi_datasetmetadata_analytics(*cpd_dm_df*, *cpd_cd_df*, *cpd_dc_df*, *spd_dm_df*, *spd_task_df*)

Analytics Description: Analyzes the DatasetMetadata.json files from both a CPD and SPD and performs a comparison check between the two. Any differences are output separately and labeled from which file it came from. The check includes:

- Program Name
- Contract Name
- Contract Number
- Contract Task or Effort Name
- Contractor Name

It also retrieves the ContractData.json from the CPD to output the Contract Definitization Date and analyzes the DatasetConfiguration.json to determine which files are reporting from Work Packages then output the results. As well as determines the total number of Tasks being reported in the SPD.

Parameters

- **cpd_dm_df** – Pandas dataframe generated from DatasetMetadata.json in a CPD
- **cpd_cd_df** – Pandas dataframe generated from ContractData.json in a CPD
- **cpd_dc_df** – Pandas dataframe generated from DatasetConfiguration.json in a CPD
- **spd_dm_df** – Pandas dataframe generated from DatasetMetadata.json in a SPD
- **spd_task_df** – Pandas dataframe generated from Tasks.json in a SPD

Returns
Pandas dataframe of joint analytics.

CSI Level 2 Tests: CPD/SPD Comparisons

ContractData-to-ProjectScheduleData Date Alignment Checks

verify_completion_dates(*cpd_cd_df*, *spd_psd_df*)

Test Description: Validates that the finish dates between the CPD and SPD align, this includes:

- Confirm that CPD.ContractData's ForecastCompletionDate aligns with SPD.ProjectScheduleData's CurrentFinishDate.

- Confirm that, if both given, CPD.ContractData's BaselineCompletionDate aligns with SPD.ProjectScheduleData's BaselineFinishDate.
- Date comparison's that are off by only 2 days or less will be considered aligned.
- For the Baseline date comparison, if the project has not been definitized, misalignments will result in a Low alert instead of a High alert.

Parameters

- **cpd_cd_df** – A Pandas' dataframe generated from cpd.ContractData.json
- **spd_psd_df** – A Pandas' dataframe generated from spd.ProjectScheduleData.json

Returns

Dictionary containing a Pandas' dataframe with any file alerts and list of message objects to display in the browser.

`verify_id_mapping`(*spd_tasks_df*, *cpd_wbs_df*, *cpd_obs_df*, *cpd_ca_df*, *cpd_wp_df*, *cpd_dc_df*)

Test Description Validates the WBSElementID, OBSElementID, ControlAccountID, and WorkPackageID columns in the spd.Tasks file are properly mapped to their respective files in the CPD submission, specifically that:

- For all Tasks that include a WBSElementID, that ID is mapped to a corresponding entry in the CPD.WBS file.
- For all Tasks that include an OBSElementID, that ID is mapped to a corresponding entry in the CPD.OBS file.
- For all Tasks that include a ControlAccountID, that ID is mapped to a corresponding entry in the CPD.ControlAccounts file.
- If the effort is in any way reporting by WorkPackages, for all Tasks that include a WorkPackageID, that ID is mapped to a corresponding entry in the CPD.WorkPackages file.

Parameters

- **spd_tasks_df** – A Pandas dataframe generated from SPD.Tasks.json.
- **cpd_wbs_df** – A Pandas dataframe generated from CPD.WBS.json.
- **cpd_obs_df** – A Pandas dataframe generated from CPD.OBS.json.
- **cpd_ca_df** – A Pandas dataframe generated from CPD.ControlAccounts.json.
- **cpd_wp_df** – A Pandas dataframe generated from CPD.WorkPackages.json.
- **cpd_dc_df** – A Pandas dataframe generated from CPD.DatasetConfiguration.json.

Returns

Dictionary containing a Pandas dataframe with any file alerts and list of message objects to display in the browser.

