Procensol
LEAD. INNOVATE. TRANSFORM.

Procensol Toolkit (PRO_TK) 3.1.0

PROCENSOL CONSULTING

Prepared by:

Phil Bye
Head of Delivery

phil.bye@procensol.com

www.procensol.com

Procensol

Procensol

Contents

1. Introduction....7
2. Dependencies and structure....8
   2.1. Regular Expression Functions....8
   2.2. Folders....8
      2.2.1. Procensol....8
      2.2.2. Procensol Toolkit Constants....8
      2.2.3. Procensol Toolkit Expressions....8
      2.2.4. Procensol Toolkit Interfaces....9
3. Expressions....10
   3.1. Validation....10
      3.1.1. PRO_TK_IsValidBankSortCodeUk....10
      3.1.2. PRO_TK_IsValidDecimalCurrency....10
      3.1.3. PRO_TK_IsValidEmailAddress....10
      3.1.4. PRO_TK_IsValidInteger....10
      3.1.5. PRO_TK_IsValidNationalInsuranceNumberUk....10
      3.1.6. PRO_TK_IsValidPostcodeUk....10
      3.1.7. PRO_TK_IsValidSafeLink....10
      3.1.8. PRO_TK_IsValidTelephoneNumberUk....10
   3.2. Formatting....10
      3.2.1. PRO_TK_FormatBoolean....10
      3.2.2. PRO_TK_FormatCurrency....11
      3.2.3. PRO_TK_FormatDate....11
      3.2.4. PRO_TK_FormatDateTime....11
      3.2.5. PRO_TK_FormatTaskAssignee....11
      3.2.6. PRO_TK_FormatTaskAssignees....11
      3.2.7. PRO_TK_FormatTaskPriority....11
      3.2.8. PRO_TK_FormatTaskStatus....11
      3.2.9. PRO_TK_FormatTime....11
      3.2.10. PRO_TK_FormatUserFirstLast....11
      3.2.11. PRO_TK_FormatUserFirstMiddleInitialLast....11
      3.2.12. PRO_TK_FormatUserFirstMiddleLast....11
      3.2.13. PRO_TK_FormatUserOrGroupFirstLast....11
      3.2.14. PRO_TK_FormatUserOrGroupFirstMiddleInitialLast....12
      3.2.15. PRO_TK_FormatUserOrGroupFirstMiddleLast....12
      3.2.16. PRO_TK_FormatText....12

3.3. Utility 12
3.3.1. PRO_TK_GenerateRecordSafeLink 12
3.3.2. PRO_TK_GenerateGUID 12
3.3.3. PRO_TK_CalculateBatchIndices 12
3.3.4. PRO_TK_CalculateFirstInstanceOfWeekday 12
3.3.5. PRO_TK_ConvertAppianDateTimeToJsonDateTime 12
3.3.6. PRO_TK_ConvertAppianDateToJsonDate 12
3.3.7. PRO_TK_ConvertJsonDateTimeToAppianDateTime 12
3.3.8. PRO_TK_ConvertJsonDateToAppianDate 13
3.3.9. PRO_TK_DisplayGroupOrUserName 13
3.3.10. PRO_TK_GetAllFieldNames 13
3.3.11. PRO_TK_GetSiteUrl 13
3.3.12. PRO_TK_GetTopLevelFieldNames 13
3.3.13. PRO_TK_Index 13
3.3.14. PRO_TK_IsNull_v2_1_1 14
3.3.15. PRO_TK_IsNull 14
3.3.16. PRO_TK_IsNullOrEmpty_v2_4_1 14
3.3.17. PRO_TK_IsNullOrEmpty 14
3.3.18. PRO_TK_IsFalse 14
3.3.19. PRO_TK_IsTrue 14
3.3.20. PRO_TK_Length_v2_4_1 14
3.3.21. PRO_TK_Length 14
3.3.22. PRO_TK_PagingInfoForAll 14
3.3.23. PRO_TK_PagingInfoForOne 14
3.3.24. PRO_TK_ParseRspQuery 14
3.3.25. PRO_TK_ReplaceNull 15
3.3.26. PRO_TK_SortBaseType 15
3.3.27. PRO_TK_GenerateDictionaryFromPath 15
3.3.28. PRO_TK_GenerateDictionaryFromListOfPaths 15
3.3.29. PRO_TK_UpdateFieldValue 15
3.3.30. PRO_TK_UpdateFieldValueAtPath 15
3.3.31. PRO_TK_GetDictionaryPaths 16
3.3.32. PRO_TK_GenerateUniqueRandomIndices 16
3.3.33. PRO_TK_RemoveRandomItemFromList 16
3.3.34. PRO_TK_RandomiseList 16
3.3.35. PRO_TK_RemoveNulls 16
3.3.36. PRO_TK_GetCurrentDateForUser 16

3.3.37. PRO_TK_GetCurrentDateTimeForUser ... 16
3.3.38. PRO_TK_CovertColumnNumberToExcelColumnName ... 17
3.3.39. PRO_TK_IsUserActive ... 17
3.3.40. PRO_TK_ConvertEpochToDateTime ... 17
3.3.41. PRO_TK_UrlEncodeString ... 17
3.3.42. PRO_TK_FindLastOccurenceOfCharacterInString ... 17
3.3.43. PRO_TK_GetFieldValueDifference ... 17
3.3.44. PRO_TK_GetUniqueArrayItems ... 17
4. Constants ... 18
4.1. Date ... 18
4.1.1. PRO_TK_FORMAT_DATE_DD_MM_YYYY ... 18
4.1.2. PRO_TK_FORMAT_DATE_DD_MMM_YYYY ... 18
4.1.3. PRO_TK_FORMAT_DATE_DDD_MMMM_YYYY ... 18
4.1.4. PRO_TK_FORMAT_DATE_MM_DD_YYYY ... 18
4.1.5. PRO_TK_FORMAT_DATE_MMM_DD_YYYY ... 18
4.1.6. PRO_TK_FORMAT_DATE_MMMM_DDD_YYYY ... 18
4.1.7. PRO_TK_FORMAT_DATE_DD_MMMM_YYYY ... 18
4.2. Time ... 18
4.2.1. PRO_TK_FORMAT_TIME_HH_MM_AM_PM ... 18
4.2.2. PRO_TK_FORMAT_TIME_HH_MM ... 18
4.2.3. PRO_TK_FORMAT_TIME_HH_MM_SS ... 18
4.2.4. PRO_TK_FORMAT_TIME_HH_MM_SS_AM_PM ... 19
4.3. Date and Time ... 19
4.3.1. PRO_TK_FORMAT_DATE_TIME_DD_MM_YYYY_HH_MM ... 19
4.3.2. PRO_TK_FORMAT_DATE_TIME_DD_MM_YYYY_HH_MM_SS ... 19
4.3.3. PRO_TK_FORMAT_DATE_TIME_DDD_MMMM_YYYY_HH_MM ... 19
4.3.4. PRO_TK_FORMAT_DATE_TIME_DDD_MMMM_YYYY_HH_MM_AM_PM ... 19
4.3.5. PRO_TK_FORMAT_DATE_TIME_DDD_MMMM_YYYY_HH_MM_SS ... 19
4.3.6. PRO_TK_FORMAT_DATE_TIME_DDD_MMMM_YYYY_HH_MM_SS_AM_PM ... 19
4.3.7. PRO_TK_FORMAT_DATE_TIME_MM_DD_YYYY_HH_MM ... 19
4.3.8. PRO_TK_FORMAT_DATE_TIME_MM_DD_YYYY_HH_MM_SS ... 19
4.3.9. PRO_TK_FORMAT_DATE_TIME_MMMM_DDD_YYYY_HH_MM_SS ... 19
4.3.10. PRO_TK_FORMAT_DATE_TIME_MMMM_DDD_YYYY_HH_MM_SS_AM_PM ... 19
4.3.11. PRO_TK_FORMAT_DATE_TIME_YYYY_MM_DD_HH_MM ... 19
4.3.12. PRO_TK_FORMAT_DATE_TIME_DD_MMMM_YYYY_HH_MM_AM_PM ... 19
4.3.13. PRO_TK_FORMAT_DATE_TIME_DD_MMMM_YYYY_HH_MM_SS_AM_PM ... 20
4.3.14. PRO_TK_FORMAT_DATE_TIME_DD_MMMM_YYYY_HH_MM_SS ... 20

4.3.15. PRO_TK_FORMAT_DATE_TIME_DD_MMMM_YYYY_HH_MM 20
4.4. Currency 20
4.4.1. PRO_TK_CURRENCY_SYMBOL_AUD 20
4.4.2. PRO_TK_CURRENCY_SYMBOL_EUR 20
4.4.3. PRO_TK_CURRENCY_SYMBOL_GBP 20
4.4.4. PRO_TK_CURRENCY_SYMBOL_USD 20
4.5. Settings 20
4.5.1. PRO_TK_SETTING_PRIMITIVE_TYPE_IDS 20
4.5.2. PRO_TK_SETTING_PRIMITIVE_TYPE_NAMES 20
4.6. Regular Expressions (RegEx) 20
4.6.1. PRO_TK_REGEX_PATTERN_EMAIL_ADDRESS 21
4.6.2. PRO_TK_REGEX_PATTERN_JSON_DATE_TIME 21
4.6.3. PRO_TK_REGEX_PATTERN_JSON_KEYS 21
4.6.4. PRO_TK_REGEX_PATTERN_POSTCODE_UK 21
4.6.5. PRO_TK_REGEX_PATTERN_SAFELINK_URL 21
4.6.6. PRO_TK_REGEX_PATTERN_TELEPHONE_NUMBER_UK 21
4.6.7. PRO_TK_REGEX_PATTERN_DECIMAL_CURRENCY 21
4.6.8. PRO_TK_REGEX_PATTERN_INTEGER 21
4.6.9. PRO_TK_REGEX_PATTERN_NATIONAL_INSURANCE_NUMBER_UK 21
4.6.10. PRO_TK_REGEX_PATTERN_BANK_SORT_CODE_UK 21
4.6.11. PRO_TK_REGEX_PATTERN_TWO_CHARACTERS_UPPERCASE_ALPHA 21
4.6.12. PRO_TK_REGEX_PATTERN_THREE_CHARACTERS_UPPERCASE_ALPHA 21
4.7. Task 22
4.7.1. PRO_TK_FORMAT_TASK_PRIORITY_IDS 22
4.7.2. PRO_TK_FORMAT_TASK_PRIORITY_NAMES 22
4.7.3. PRO_TK_FORMAT_TASK_STATUS_IDS 22
4.7.4. PRO_TK_FORMAT_TASK_STATUS_NAMES 22
4.8. Records 22
4.8.1. PRO_TK_RECORD_USER 22
5. Interfaces 23
5.1.1. PRO_TK_DecimalField 23
6. Further notes 24
6.1. Security 24
6.2. Credits 24

1. Introduction

Procensol were the first Appian Partner in the UK and have been working with Appian for well over a decade. This toolkit is a result of years of project experience, with each expression rule or constant representing a distinct piece of functionality that proved to be a common requirement across multiple projects.

The toolkit includes a lot of useful functionality, such as:

- Task assignee formatting
- JSON date conversion
- Recursive index functionality
- Retrieve CDT field names
- Sort base types
- Update CDT or dictionary fields
- Common regular expressions
- Common date and time formats

If you have any suggestions for new functionality or find any bugs, let Neil Coppin or Phil Bye know and they will try to help!

2. Dependencies and structure

2.1. Regular Expression Functions

Some aspects of the toolkit rely upon functionality provided by the Regular Expression Functions plugin, available at the link shown. The plugin is cloud-approved and simple to deploy.

Note that 1.x.x versions of the toolkit rely on version 1.0.0 of the Regular Expression Functions plugin, but 2.x.x versions of the toolkit rely on version 2.1.0 of the same plugin.

2.2. Folders

The rules, constants and interfaces that make up the toolkit are logically divided into folders where appropriate.

2.2.1. Procensol

This folder is the top-level rules folder for Procensol and is common across all Procensol applications.

2.2.1.1. Procensol Toolkit

This is the top-level rules folder for the Procensol Toolkit application. All other rules folders sit below this.

2.2.2. Procensol Toolkit Constants

The top-level constants folder for the Procensol Toolkit application

2.2.2.1. Procensol Toolkit Constants – Formatting

A folder containing constants for the Procensol Toolkit that are to be used for formatting.

2.2.2.2. Procensol Toolkit Constants – Records

A folder containing constants related to records.

2.2.2.3. Procensol Toolkit Constants - Regular Expression

A folder containing constants for the Procensol Toolkit that hold regular expressions.

2.2.2.4. Procensol Toolkit Constants – Settings

A folder containing constants that provide settings for the procensol toolkit.

2.2.3. Procensol Toolkit Expressions

The top-level expressions folder for the Procensol Toolkit application.

2.2.3.1. Procensol Toolkit Expressions – Formatting

A folder containing expressions for the Procensol Toolkit that are to be used for formatting.

2.2.3.2. Procensol Toolkit Expressions – Utility

A folder containing expressions for the Procensol Toolkit for general purpose and utility use.

2.2.3.3. Procensol Toolkit Expressions – Validation

A folder containing expressions for the Procensol Toolkit that are to be used for validation.

2.2.4. Procensol Toolkit Interfaces

A folder containing interfaces for the Procensol Toolkit.

3. Expressions

3.1. Validation

These validation expressions generally return true or false if the provided value or values match the conditions defined by the rule. They are particularly useful when used in the validations property of an interface component.

3.1.1. PRO_TK_IsValidBankSortCodeUk

Returns true if the provided value matches a defined bank sort code syntax, or false if it does not.

3.1.2. PRO_TK_IsValidDecimalCurrency

Returns true if the provided value matches a defined decimal currency syntax, or false if it does not.

3.1.3. PRO_TK_IsValidEmailAddress

Returns true if the provided value matches a defined email address syntax, or false if it does not.

3.1.4. PRO_TK_IsValidInteger

Returns true if the provided value matches an integer, or false if it does not.

3.1.5. PRO_TK_IsValidNationalInsuranceNumberUk

Returns true if the provided value matches a defined UK national insurance number syntax, or false if it does not.

3.1.6. PRO_TK_IsValidPostcodeUk

Returns true if the provided value matches a defined UK postcode syntax, or false if it does not.

3.1.7. PRO_TK_IsValidSafeLink

Returns true if the provided value matches a defined safe link syntax, or false if it does not.

3.1.8. PRO_TK_IsValidTelephoneNumberUk

Returns true if the provided value matches a defined UK phone number syntax, or false if it does not.

3.2. Formatting

These reusable formatting expressions make formatting various types of data much easier and will contribute towards consistent user interfaces.

3.2.1. PRO_TK_FormatBoolean

Accepts a boolean and formats it as the appropriate status icon. This is most useful when displaying a Boolean in a!gridField().

3.2.2. PRO_TK_FormatCurrency

Returns the amount provided in the provided currency format. The following constants are provided by the toolkit for a selection of commonly used currencies.

Currency (Required):

Dollar = cons!PRO_TK_CURRENCY_SYMBOL_AUD / cons!PRO_TK_CURRENCY_SYMBOL_USD,

Euro = cons!PRO_TK_CURRENCY_SYMBOL_EUR,

Pound = cons!PRO_TK_CURRENCY_SYMBOL_GBP.

3.2.3. PRO_TK_FormatDate

Accepts a date input and a text format and applies that format using the text() function.

3.2.4. PRO_TK_FormatDateTime

Accepts a date-time input and a text format and applies that format using the text() function.

3.2.5. PRO_TK_FormatTaskAssignee

Accepts a single assignee and returns a formatted group and/or user name as appropriate.

3.2.6. PRO_TK_FormatTaskAssignees

Accepts a list of assignees and returns a string of formatted group and/or user names as appropriate, separated by commas and spaces as required. This function is particularly useful when used to format the assignees value that is returned by a task or process report

3.2.7. PRO_TK_FormatTaskPriority

Accepts a task priority ID input and returns the name for that task priority.

3.2.8. PRO_TK_FormatTaskStatus

Accepts a task status ID input and returns the name for that task status.

3.2.9. PRO_TK_FormatTime

Accepts a time input and a text format and applies that format using the text() function.

3.2.10. PRO_TK_FormatUserFirstLast

Accepts a user input and returns the user's name in the format "First Last".

3.2.11. PRO_TK_FormatUserFirstMiddleInitialLast

Accepts a user input and returns the user's name in the format "First M Last".

3.2.12. PRO_TK_FormatUserFirstMiddleLast

Accepts a user input and returns the user's name in the format "First Middle Last".

3.2.13. PRO_TK_FormatUserOrGroupFirstLast

Accepts a user or group input and formats to provide the user's first and last name or group name.

3.2.14. PRO_TK_FormatUserOrGroupFirstMiddleInitialLast

Accepts a user or group input and returns the user's name in the format "First M Last" or group name.

3.2.15. PRO_TK_FormatUserOrGroupFirstMiddleLast

Accepts a user or group input and returns the user's name in the format "First Middle Last" or group name.

3.2.16. PRO_TK_FormatText

Executes the text() function, but handles null values by returning the original value.

3.3. Utility

These utility expressions provide reusable functionality that might otherwise be technically difficult or long-winded to achieve in-line within an expression or interface.

3.3.1. PRO_TK_GenerateRecordSafeLink

Returns a a!safeLink() with a uri that will open a record in a new tab. The view parameter can be used to navigate to a specific record view, whilst the site and page identifiers can be used to generate a link that will open within a site.

3.3.2. PRO_TK_GenerateGUID

Returns a Globally Unique Identifier. Note that this rule can safely be used within a!forEach to generate unique identifiers per iteration.

3.3.3. PRO_TK_CalculateBatchIndices

Calculates the next set of indices for a batch when given a set of indices, an array and a batch size. The indices parameter should be left blank for the first iteration, and the previously calculated set of indices should be passed back into the rule fir each iteration.

3.3.4. PRO_TK_CalculateFirstInstanceOfWeekday

Returns the date that represents the first occurrence of a specified weekday for a month/year combination. Note that returnType controls how the days of the week are numbered as per the weekday() function. '1' correlates 1 ... 7 with Sunday ... Saturday (this is the default). '2' correlates 1 ... 7 with Monday ... Sunday. '3' correlates 0 ... 6 with Monday ... Sunday.

3.3.5. PRO_TK_ConvertAppianDateTimeToJsonDateTime

Accepts an Appian datetime and converts it to a JSON datetime string.

3.3.6. PRO_TK_ConvertAppianDateToJsonDate

Accepts an Appian date and converts it to a JSON date string.

3.3.7. PRO_TK_ConvertJsonDateTimeToAppianDateTime

Accepts a JSON formatted Datetime and returns an Appian date time.

3.3.8. PRO_TK_ConvertJsonDateToAppianDate

Accepts a JSON formatted date and returns an Appian date.

3.3.9. PRO_TK_DisplayGroupOrUserName

Get display for user or group based on runtime type

3.3.10. PRO_TK_GetAllFieldNames

Returns all attribute or key names for a dictionary or CDT. Note that the attributes will include those that are contained within child values of the dictionary or CDT.

3.3.11. PRO_TK_GetSiteUrl

Returns the URL for the site.

3.3.12. PRO_TK_GetTopLevelFieldNames

Returns the top-level attribute or key names for a dictionary or CDT.

Note that, due to the way regular expressions are applied in Java, this expression may run slowly if a large amount of data is passed in. If this expression is to be used with a large amount of data, we recommend that the logic below be replaced with a call to fn!getcdtproperties(), which is available from the Appian Appmarket.

3.3.12.1. PRO_TK_GetTopLevelFieldNamesHelper

A helper rule used by PRO_TK_GetTopLevelFieldNames in order to avoid use of a!forEach, as a!forEach cannot be used by a rule that is called by fn!reduce.

Note that if the logic within PRO_TK_GetTopLevelFieldNames has been replaced with a call to fn!getcdtproperties(), this rule will now be redundant.

3.3.13. PRO_TK_Index

An enhanced version of the index function that accepts a dictionary and a dot-notated field list (eg case.data.process.status.name), and extracts the field defined by the dot notation.

The dictionary input can be a CDT (including nested elements) or a dictionary object (eg a result from a!fromJson().

This rule now handles square-bracket notation for accessing array elements at any level. For example, indexing using paths such as this will now return the referenced value:

```
data.node.array[3]
```

Accessing values at multiple levels of array will also work:

```
data.node[58].array[3].value
```

If a value doesn't exist or a path is not valid, an empty text value will be returned, as it was before. The rule is backward compatible, so no changes are required for existing expressions.

3.3.13.1. PRO_TK_IndexHelper

A helper rule used by PRO_TK_Index in order to avoid use of a!forEach as in some Appian versions, a!forEach cannot be used by a rule that is called by fn!reduce. Additionally, a!forEach does not perform as well as fn!apply.

3.3.14. PRO_TK_IsNull_v2_1_1

DEPRECATED - Returns true if the input value is null, or false if it is not, also returns true if passed an empty array

3.3.15. PRO_TK_IsNull

Returns true if the input value is null, or false if it is not.

3.3.16. PRO_TK_IsNullOrEmpty_v2_4_1

DEPRECATED - Returns true if value is null or an empty array. Note that, if the array contains only null values, this rule will return false.

3.3.17. PRO_TK_IsNullOrEmpty

Returns true if value is null or an empty array. Equivalent to a!IsNullOrEmpty().

3.3.18. PRO_TK_IsFalse

Returns true for all falsey statements, true otherwise.

3.3.19. PRO_TK_IsTrue

Returns true for all truthy statements, false otherwise

3.3.20. PRO_TK_Length_v2_4_1

DEPRECATED - Returns 0 if the input value is null, or length if it is not. Note that arrays that contain only null items will return a length of zero.

3.3.21. PRO_TK_Length

Returns 0 if the input value is null, or the length of the input value if it is not null.

3.3.22. PRO_TK_PagingInfoForAll

Returns a paging info object with a start index of 1, a batch size of -1 and an optional sort order.

3.3.23. PRO_TK_PagingInfoForOne

Returns a paging info object with a start index of 1, a batch size of 1 and an optional sort order.

3.3.24. PRO_TK_ParseRspQuery

For use with expression-backed records; parses an rsp!query object and returns a dictionary. Note that as of 18.3, this rule shouldn't really be required.

3.3.25. PRO_TK_ReplaceNull

Returns nullableValue only if it is not null. Else return replacementValue.

3.3.26. PRO_TK_SortBaseType

Sorts and returns an array of base type (e.g. string, integer etc) in ascending order, or descending if ascending is set to false.

3.3.27. PRO_TK_GenerateDictionaryFromPath

Accepts a dot-notated path and an optional value and generates a dictionary equivalent to that path. If a value is not provided, the value of the lowest level node will be null.

3.3.27.1. PRO_TK_GenerateDictionaryFromPathHelper

A helper rule used by PRO_TK_GenerateDictionaryFromPath in order to avoid use of a!forEach, as a!forEach cannot be used by a rule that is called by fn!reduce.

3.3.28. PRO_TK_GenerateDictionaryFromListOfPaths

Accepts a list of dot-notated paths and an optional default value and generates a dictionary equivalent to the list of paths.

Note that if no paths are provided, an empty dictionary will be returned and, if a default value is not provided, the value of each lowest level node will be null.

3.3.29. PRO_TK_UpdateFieldValue

Note: As of Appian version 21.2, the functionality provided by this rule can be more easily achieved using a!update().

Accepts a field name, field value, and a dictionary or CDT and either updates the appropriate field within the dictionary or CDT to the given value or adds the field with the given value if the field does not exist.

Certain types of new value - text, boolean, dictionary and integer for example - are handled well but dates and times are not. This is because the expression converts to and from JSON, which can result in unexpected formatting of the new value in the dictionary.

Note that whilst this rule appears to accept an array of CDT or dictionary, it cannot perform an update for a field in a list of dictionary or CDT and will fail.

3.3.29.1. PRO_TK_UpdateFieldValueHelper

A helper rule used by PRO_TK_UpdateFieldValue in order to avoid use of a!forEach, as a!forEach cannot be used by a rule that is called by fn!reduce.

3.3.30. PRO_TK_UpdateFieldValueAtPath

Accepts a dot-notated path, a value, and a dictionary or CDT and either updates the appropriate field within the dictionary or CDT to the given value or adds the field (or dictionary) with the given value if the field does not exist.

If a field already exists and is a primitive type (text, for example), attempting to add a field to that existing field will fail. Certain types of new value - text, boolean, dictionary and integer for example - are handled

well but dates and times are not. This is because the expression converts to and from JSON, which can result in unexpected formatting of the new value in the dictionary.

Note that whilst this rule appears to accept an array of CDT or dictionary, it cannot perform an update for a field in a list of dictionary or CDT and will fail. Also note that repeated paths at multiple levels within a dictionary can cause unexpected behaviour, such as the wrong node being updated with the given value.

3.3.31. PRO_TK_GetDictionaryPaths

Accepts a CDT or Dictionary and a list of paths and returns a Dictionary containing only the paths requested.

This rule is particularly useful when working with responses from web services, where a sizable response is received from the service and converted to a dictionary, but only a small part of that response is required by Appian.

Retaining the whole response is often not desirable as very large responses can cause increased memory usage, which can occasionally lead to performance issues.

3.3.32. PRO_TK_GenerateUniqueRandomIndices

Accepts a range and a length and returns a list (of the specified length) of unique random indices between 1 and the given range.

Useful for generating random samples of data - for example, to select 25 of 100 items, pass a length of 25 and a range of 100, and this rule will generate 25 random indices between 1 and 100.

Note: renamed from PRO_TK_GenerateUniqueRandomNumberList in Toolkit version 2.3.0.

3.3.33. PRO_TK_RemoveRandomItemFromList

Removes a single random item from a given list.

Note that the index input is only present for use with reduce() and should be ignored if the rule is used anywhere other than PRO_TK_GenerateUniqueRandomNumberList.

3.3.34. PRO_TK_RandomiseList

Accepts a list of items and returns the same list in a randomised order.

3.3.35. PRO_TK_RemoveNulls

Removes null values from an array. Note that, if a non-array value is passed in, this expression will return an error.

3.3.36. PRO_TK_GetCurrentDateForUser

Returns the current date, localised using the time zone for either a provided user or, if the provided user is null, the time zone for the logged in user.

3.3.37. PRO_TK_GetCurrentDateTimeForUser

Returns the current dateTime, localised using the time zone for either a provided user or, if the provided user is null, the time zone for the logged in user.

3.3.38. PRO_TK_CovertColumnNumberToExcelColumnName

Accepts a column number and returns the excel column name (e.g. A, F, AD, CED etc) that is equivalent to the column number.

Note that this rule will perform incrementally more slowly if column numbers greater than 26 or 702 are passed in, as this means more possible column names must be generated. However, if you're trying to find the 800th column name in excel, you're already doing something weird and this shouldn't be much of a surprise.

3.3.39. PRO_TK_IsUserActive

Returns true if the provided username belongs to an active user, or false if it does not.

3.3.40. PRO_TK_ConvertEpochToDateTime

Accepts an Epoch timestamp in milliseconds and converts it to a date and time type. Note that the epoch input is optional, and will default to 01/01/1970 if it is not provided.

3.3.41. PRO_TK_UrlEncodeString

Encodes a string for use in a URL, replacing non-ASCII or disallowed characters (such as spaces or double quotes) with compatible characters (such as %20 or %22).

3.3.42. PRO_TK_FindLastOccurenceOfCharacterInString

Returns the position of the last occurrence of a given character in a string, or zero if it does not occur.

3.3.43. PRO_TK_GetFieldValueDifference

Returns an array of map containing the differences between two Dictionaries, Maps, CDTs, or Records. Note that the types must match, and arrays are not supported; use this rule within a loop if the differences between arrays are required.

3.3.44. PRO_TK_GetUniqueArrayItems

Returns the unique items from an array or arrays. Similar to (and uses) fn!union, but can handle multiple arrays at once.

4. Constants

4.1. Date

These constants contain formatting patterns for commonly used date formats that can be used in conjunction with the included date format rules when displaying values on-screen.

4.1.1. PRO_TK_FORMAT_DATE_DD_MM_YYYY

A date format of dd/MM/yyyy

4.1.2. PRO_TK_FORMAT_DATE_DD_MMM_YYYY

A date format of dd MMM YYYY

4.1.3. PRO_TK_FORMAT_DATE_DDD_MMMM_YYYY

A date format of ddd MMMM yyyy

4.1.4. PRO_TK_FORMAT_DATE_MM_DD_YYYY

A date format of MM/dd/yyyy

4.1.5. PRO_TK_FORMAT_DATE_MMM_DD_YYYY

A date format of dd MMM YYYY

4.1.6. PRO_TK_FORMAT_DATE_MMMM_DDD_YYYY

A date format of MMMM ddd yyyy

4.1.7. PRO_TK_FORMAT_DATE_DD_MMMM_YYYY

A date format of dd MMMM YYYY

4.2. Time

These constants contain formatting patterns for commonly used time formats that can be used in conjunction with the included time format rules when displaying values on-screen.

4.2.1. PRO_TK_FORMAT_TIME_HH_MM_AM_PM

A time format of HH:mm

4.2.2. PRO_TK_FORMAT_TIME_HH_MM

A time format of HH:mm AM/PM

4.2.3. PRO_TK_FORMAT_TIME_HH_MM_SS

A time format of HH:mm:ss

4.2.4. PRO_TK_FORMAT_TIME_HH_MM_SS_AM_PM

A time format of HH:mm:ss AM/PM

4.3. Date and Time

These constants contain formatting patterns for commonly used date and time formats that can be used in conjunction with the included date and time format rules when displaying values on-screen.

4.3.1. PRO_TK_FORMAT_DATE_TIME_DD_MM_YYYY_HH_MM

A date time format of dd/MM/yyyy HH:mm

4.3.2. PRO_TK_FORMAT_DATE_TIME_DD_MM_YYYY_HH_MM_SS

A date time format of dd/MM/yyyy HH:mm:ss

4.3.3. PRO_TK_FORMAT_DATE_TIME_DDD_MMMM_YYYY_HH_MM

A date time format of ddd MMMM yyyy HH:mm

4.3.4. PRO_TK_FORMAT_DATE_TIME_DDD_MMMM_YYYY_HH_MM_AM_PM

A date time format of ddd MMMM yyyy HH:mm AM/PM

4.3.5. PRO_TK_FORMAT_DATE_TIME_DDD_MMMM_YYYY_HH_MM_SS

A date time format of ddd MMMM yyyy HH:mm:ss

4.3.6. PRO_TK_FORMAT_DATE_TIME_DDD_MMMM_YYYY_HH_MM_SS_AM_PM

A date time format of ddd MMMM yyyy HH:mm:ss AM/PM

4.3.7. PRO_TK_FORMAT_DATE_TIME_MM_DD_YYYY_HH_MM

A date time format of MM/dd/yyyy HH:mm

4.3.8. PRO_TK_FORMAT_DATE_TIME_MM_DD_YYYY_HH_MM_SS

A date time format of MM/dd/yyyy HH:mm:ss

4.3.9. PRO_TK_FORMAT_DATE_TIME_MMMM_DDD_YYYY_HH_MM_SS

A date time format of MMMM ddd yyyy HH:mm:ss

4.3.10. PRO_TK_FORMAT_DATE_TIME_MMMM_DDD_YYYY_HH_MM_SS_AM_PM

A date time format of MMMM ddd yyyy HH:mm:ss AM/PM

4.3.11. PRO_TK_FORMAT_DATE_TIME_YYYY_MM_DD_HH_MM

A date time format of yyyy-MM-dd HH:mm

4.3.12. PRO_TK_FORMAT_DATE_TIME_DD_MMMM_YYYY_HH_MM_AM_PM

A date time format of dd MMMM yyyy HH:mm AM/PM

4.3.13. PRO_TK_FORMAT_DATE_TIME_DD_MMMM_YYYY_HH_MM_SS_AM_PM

A date time format of dd MMMM yyyy HH:mm:ss AM/PM

4.3.14. PRO_TK_FORMAT_DATE_TIME_DD_MMMM_YYYY_HH_MM_SS

A date time format of dd MMMM yyyy HH:mm:ss

4.3.15. PRO_TK_FORMAT_DATE_TIME_DD_MMMM_YYYY_HH_MM

A date time format of dd MMMM yyyy HH:mm

4.4. Currency

These constants contain currency symbols for commonly used currencies. The included constants are not intended to be an exhaustive list and will be added to over time.

4.4.1. PRO_TK_CURRENCY_SYMBOL_AUD

The currency symbol for the Australian Dollar.

4.4.2. PRO_TK_CURRENCY_SYMBOL_EUR

The currency symbol for the Euro.

4.4.3. PRO_TK_CURRENCY_SYMBOL_GBP

The currency symbol for the British Pound.

4.4.4. PRO_TK_CURRENCY_SYMBOL_USD

The currency symbol for the US Dollar.

4.5. Settings

These constants contain system values that are used when comparing or displaying system types or values.

4.5.1. PRO_TK_SETTING_PRIMITIVE_TYPE_IDS

A list of IDs for the system primitive types. This list may not be exhaustive and will be added to as and when required.

4.5.2. PRO_TK_SETTING_PRIMITIVE_TYPE_NAMES

A list of names for the system primitive types. This list may not be exhaustive and will be added to as and when required.

4.6. Regular Expressions (RegEx)

These constants contain Regular Expression patterns and can be used in conjunction with the Regular Expression Functions to perform complex pattern matching for common validation requirements.

4.6.1. PRO_TK_REGEX_PATTERN_EMAIL_ADDRESS

A regular expression pattern for validating the format of an email address.

4.6.2. PRO_TK_REGEX_PATTERN_JSON_DATE_TIME

A regular expression pattern for validating the format of a JSON datetime. Note that this regex is explicitly not the ISO8601 format, as some variants of that format won't match this pattern.

4.6.3. PRO_TK_REGEX_PATTERN_JSON_KEYS

A regex pattern that matches any key names within JSON.

4.6.4. PRO_TK_REGEX_PATTERN_POSTCODE_UK

A regular expression pattern for validating the format of a UK postcode.

4.6.5. PRO_TK_REGEX_PATTERN_SAFELINK_URL

A regular expression pattern for validating URLs to be used as safelinks.

4.6.6. PRO_TK_REGEX_PATTERN_TELEPHONE_NUMBER_UK

A regular expression pattern for validating the format of a UK telephone number.

4.6.7. PRO_TK_REGEX_PATTERN_DECIMAL_CURRENCY

A regular expression pattern for validating the format of a decimal value with two decimal places - specifically intended for currency. Note that negative values are permitted by this regular expression and the two decimal places are required (i.e. 1.0 would not pass validation).

4.6.8. PRO_TK_REGEX_PATTERN_INTEGER

A regular expression pattern for validating the format of an integer. Note that negative values are permitted by this regular expression.

4.6.9. PRO_TK_REGEX_PATTERN_NATIONAL_INSURANCE_NUMBER_UK

A regular expression pattern for validating the format of a UK National Insurance number.

4.6.10. PRO_TK_REGEX_PATTERN_BANK_SORT_CODE_UK

A regular expression pattern for validating the format of a UK bank sort code.

4.6.11. PRO_TK_REGEX_PATTERN_TWO_CHARACTERS_UPPERCASE_ALPHA

A regular expression pattern for validating the format of a string that should contain an uppercase, two-letter code, e.g. "AB". This is useful for validating things such as two-letter country codes.

4.6.12. PRO_TK_REGEX_PATTERN_THREE_CHARACTERS_UPPERCASE_ALPHA

A regular expression pattern for validating the format of a string that should contain an uppercase, three-letter code, e.g. "AB". This is useful for validating things such as three-letter country codes.

4.7. Task

These constants contain values that can be used when formatting task data that has been returned from task and process reports.

4.7.1. PRO_TK_FORMAT_TASK_PRIORITY_IDS

The list of possible task priority IDs for use when interpreting the priority of a task.

4.7.2. PRO_TK_FORMAT_TASK_PRIORITY_NAMES

The list of possible task priority names for use when interpreting the priority of a task.

4.7.3. PRO_TK_FORMAT_TASK_STATUS_IDS

The list of possible task status IDs for use when interpreting the status of a task.

4.7.4. PRO_TK_FORMAT_TASK_STATUS_NAMES

The list of possible task status names for use when interpreting the status of a task.

4.8. Records

These constants contain values that refer to records within the system.

4.8.1. PRO_TK_RECORD_USER

The user record. Note that the user record is a system-generated record to which limited changes can be made.

5. Interfaces

5.1.1. PRO_TK_DecimalField

Returns a text field that functions as a decimal field but that displays the value with formatting and optional currency.

6. Further notes

6.1. Security

The default security for all items within the toolkit is set to viewer for all other users.

6.2. Credits

This toolkit is a result of many people, working on many projects and across many years of experience. So – big thanks to…

- The creators of the original Common Objects (APN_) application
- Tharshula Selvarasa for putting together the GUID functionality
- Ben Nicholls for actually reading the descriptions and spotting required corrections
- Suresh Gunawardane for the record URI generation rules
- Jansen NG for his original JSON date time work

If someone has been missed out, it's definitely not deliberate – let Phil Bye know!

Procensol
LEAD. INNOVATE. TRANSFORM.

Europe

Waterloo House
20 Waterloo Road
Birmingham
B2 5TB
United Kingdom
Tel: +44(0) 121 231 7045
E: info@procensol.com

APAC

Level 2
199 George Street
Brisbane
Queensland 4000
Australia
Tel: +61 (0)7 3012 6634
E: info@procensol.com.au

Collins Street Tower
Level 3
480 Collins St
Melbourne
Victoria 3000
Australia
Tel: +61 (0)3 8610 6893