# **Proposal:**

**Summary**

* Add a new implicit conversion to clarify the operational semantics of *date* and *date and time* equivalence
* Fix a typo

**Details**

The changes are baselined on DMN 1.3 **dtc-19-12-06.pdf**.

**Position:**

**Page 125 Section 10.3.2.3.5 date**

**Replace**

Where necessary, including the value*dt*function (see 10.3.2.3.6), a date value is considered to be equivalent to a date time value in which the time of day is UTC midnight (00:00:00).

**with**

When a date value is subject to implicit conversions (10.3.2.9.4) it is considered to be equivalent to a date time value in which the time of day is UTC midnight (00:00:00).

**Position:**

**Page 134 Section 10.3.2.9.4 Type conversions**

**Replace**

There are several possible type conversions:

- *to singleton list*:

When the type of the expression is T and the target type is List<T> the expression is converted to a singleton list.

- *from singleton list*:

When the type of the expression is List<T>, the value of the expression is a singleton list and the target type is T, the expression is converted by unwraping the first element.

- *conforms to:*

When the type of the expression is T1, the target type is T2, and T1 conforms to T2 the value of expression remains unchanged. Otherwise the result is **null**.

There are several kinds of contexts in which implicit conversions may occur:

* Filter context (10.3.2.5) in which a filter expression is present. The expression to be filtered is subject to implicit conversion *to singleton list*.
* Invocation context (Table 63) in which an argument is bound to a formal parameter of a function. The arguments are subject to implicit conversion *from singleton list*.
* Binding contexts in which the value of an expression is bound to a variable with associated type information (e.g. binding actual parameters to formal parameters in an invocation, or binding the result of a decision’s logic to the decsion’s output variable). The expression is subject to *conforms to* conversion.

**with**

In implicit type conversions, the data type is converted automatically without loss of information. There are several possible implicit type conversions:

- *to singleton list*:

When the type of the expression is T and the target type is List<T> the expression is converted to a singleton list.

- *from singleton list*:

When the type of the expression is List<T>, the value of the expression is a singleton list and the target type is T, the expression is converted by unwraping the first element.

- *to midnight conversion*:

When the type of the expression is date and the target type is date and time, the expression is converted to a date time value in which the time of day is UTC midnight (00:00:00).

There is one type of conversion to handle semantic errors:

- *conforms to:*

When the type of the expression is T1, the target type is T2, and T1 conforms to T2 the value of expression remains unchanged. Otherwise the result is **null**.

There are several kinds of contexts in which conversions may occur:

* Filter context (10.3.2.5) in which a filter expression is present. The expression to be filtered is subject to implicit conversion *to singleton list*.
* Invocation context (Table 63) in which an actual parameter is bound to a formal parameter of a function. The actual parameter is subject to implicit conversions.
* Binding contexts in which the result of a DRG Element’s logic is bound to the output variable. If after applying the implicit conversions the converted value and the target type do not conform, the *conforms to* conversion is applied.