



# Kowari Documentation

S Y M B O L I C  
T R A N S F O R M A T I O N   F I X  
S T A G E   1

netymon pty ltd

Unit 3/13 Denham St Annerley 4103 Australia - email: [mail@netymon.com](mailto:mail@netymon.com) – acn: 113 088 145

# Motivation

The following Query fails prior to this fix:

```
create <rmi://localhost/server1#foo> ;  
insert <test:1> <http://tucana.org/tucana#lt> '4.0'  
    into <rmi://localhost/server1#foo> ;  
select $s from <rmi://localhost/server1#foo> where  
    $s <http://tucana.org/tucana#lt> '4.0' ;
```

The expected result is naturally <test:1>; due to the current IntervalResolver using predicates to identify constraints to collapse into intervals the result is all doubles in the stringpool less than '4.0'. When we look to fix this problem we find that there is no reasonable way for a SymbolicTransformation to determine the ModelType of a given constraint, and consequently it's applicability to the given transform.

## Summary of Solution

The solution is to provide explicit access to a constraint's model from via the Constraint interface; and to modify the SymbolicTransformation interface to provide a context parameter that can be used to map a model URI obtained from the constraint to a model-type URI. With the use of an explicit model within a Constraint we break the previously implicit association between element-3 of a constraint and the model used by LocalQueryResolver. Consequently we also need to update LocalQueryResolver::resolve(Constraint, ModelResource) to reflect this. This requires adding support for explicit localisation of Constraints into ConstraintOperations.

This along with associated test cases has been done, and is available as Revision 16 at <http://svn.netymon.com/public/netpr/>.

# Unit Tests

The fix is validated via two additional unit tests in IntervalTransformationUnitTest, and an three additional jxunit tests under jxdata/iTql/transformpreserve.

## IntervalTransformationUnitTest

### ::test6IgnoreNonXSD

transform conjunction of two unrelated constraints and ensure result is null.

### ::test7PreserveNonXSD

transform conjunction of four constraints. Two unrelated, two forming an bounded interval. Result should be the conjunction of the two unrelated constraints with a bounded interval constraint.

## Jxunit – transformpreserve

Three tests that validate the test query given in the motivation.

# Implementation Comments

See subversion logs at <http://svn.netymon.com/public/netpr> for specific details of files changed and how. Several comments are worth making:

IntervalConstraint had been a ConstraintExpression, it should be a Constraint, and has been made so. It also naturally needed to have a getModel() method provided, as did every other Constraint in the system.

All existing calls to getElement(3) have been changed to call getModel(). The change is trivial and idempotent, but it has affected every Resolver in Kowari.

A new interface SymbolicTransformationContext has been added to define the methods made available to transformations. At the moment it is implemented by DatabaseOperationContext, this will likely change in the future, but that change should not affect transformations.

The new method to map a model URI to a model type URI is SymbolicTransformationContext.mapToModelTypeURI().

IntervalTransformation has been updated to call getModel() and to check this using the context against the xsd-model-type-uri provided by the XSDResolverFactory.

A new interface ConstraintLocalization has been added. This will

eventually need to be made available via `ConstraintDescriptor`, and will be in the next stage of the fix. For the moment it is pre-initialised with localisations for `ConstraintImpl` and `ConstraintNegation` in `LocalQueryResolver`. This is accompanied by related functions in `ConstraintOperations`.

`LocalQueryResolver.resolve` has been greatly simplified by the use of `ConstraintOperations.localize()`.

`DatabaseOperationContext` naturally has a new method, `mapToModelTypeURI()` and is now passed by `DatabaseSession` to `SymbolicTransformations`.