

Compiler – Runtime Contract

Authors: Dan, Dana, David, Donald, Markos, Tim, Vinayak

This document contains a list of agreements between the compiler and the runtime system (i.e., the set of iterators).

- **Arithmetics, Comparisons:** The information should pass (if known) the static types of the operands (for numerics) and target to the runtime iterator. There is only one iterator for each arithmetic operation and kind of comparison. Passing the concrete static types will save cost at runtime for type promotions and to detect the right target type.
- **Normal function calls:** Compiler generates code for `FuncParamCast`, if necessary. Iterators assume that the input is in the right type and need not check for that.
- **FuncParamCast:** There is a separate iterator for each step of a `FuncParamCast` (e.g., atomization, check occurrence, check „isSubTypeOf“). Compiler will generate code only for those steps that are needed at runtime.
- **Path Expressions:** Each step is implemented by a separate iterator. There is no „super-XPath“ iterator. Match is a separate iterator.
- **FLWOR:** Corresponding flags (e.g., materialization of Let, Variable bindings, etc.) will be specified by Tim.
- To be continued....