

```

(*****
(* HaskCore: basically GHC's CoreSyn.Expr imported into Coqland *)
(*****)

Generalizable All Variables.
Require Import Preamble.
Require Import General.
Require Import Coq.Strings.String.
Require Import HaskKinds.
Require Import HaskLiteralsAndTyCons.
Require Import HaskCoreTypes.
Require Import HaskCoreVars.

(* this type must extract to EXACTLY match TypeRep.Type *)
Inductive CoreExpr {b:Type} :=
| CoreEVar   : CoreVar           -> CoreExpr
| CoreELit   : HaskLiteral       -> CoreExpr
| CoreEApp   : CoreExpr          -> CoreExpr -> CoreExpr
| CoreELam   : b                 -> CoreExpr -> CoreExpr
| CoreELet   : CoreBind          -> CoreExpr -> CoreExpr
| CoreECase  : CoreExpr -> b -> CoreType -> list (@triple CoreAltCon (list b) CoreExpr) -> CoreExpr
| CoreECast  : CoreExpr          -> CoreCoercion -> CoreExpr
| CoreENote  : Note              -> CoreExpr -> CoreExpr
| CoreEType  : CoreType          -> CoreExpr
with CoreBind {b:Type} :=
| CoreNonRec : b -> CoreExpr -> CoreBind
| CoreRec    : list (b * CoreExpr) -> CoreBind.
Extract Inductive CoreExpr =>
  "CoreSyn.Expr" [
    "CoreSyn.Var"
    "CoreSyn.Lit"
    "CoreSyn.App"
    "CoreSyn.Lam"
    "CoreSyn.Let"
    "CoreSyn.Case"
    "CoreSyn.Cast"
    "CoreSyn.Note"
    "CoreSyn.Type" ].
Extract Inductive CoreBind =>
  "CoreSyn.Bind" [ "CoreSyn.NonRec" "CoreSyn.Rec" ].

```

```
Variable coreExprToString : @CoreExpr CoreVar -> string. Extract Inlined Constant coreExprToString => "outputableToString".
Instance CoreExprToString : ToString (@CoreExpr CoreVar) := { toString := coreExprToString }.

Variable coreTypeOfCoreExpr : @CoreExpr CoreVar -> CoreType.
  Extract Inlined Constant coreTypeOfCoreExpr => "CoreUtils.exprType".
```