```
{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w#}::Indices(position=independent).

\nabla{#}::Derivative.
\partial{#}::PartialDerivative.

Dx{#}::LaTeXForm("{\Dx}").  # LCB: This used to cause a bug, it kills ::KeepWeight for Dx
                            # LCB: But I'm not sure that this is still true.
                            # LCB: So just to be safe I'll leave all Cadabra sources unchanged.
                            # LCB: I can use this trick in this file because it doesn't use ::KeepWeight
import cdblib

# ------------------------------------------------------------------------------
# the metric

# metric in terms of R and \partial R
tmp = cdblib.get ('g_ab_3','metric.export')         # cdb (metric3,tmp)
tmp = cdblib.get ('g_ab_4','metric.export')         # cdb (metric4,tmp)
tmp = cdblib.get ('g_ab_5','metric.export')         # cdb (metric5,tmp)
tmp = cdblib.get ('g_ab_6','metric.export')         # cdb (metric6,tmp)

# metric in terms of R and \nabla R
tmp = cdblib.get ('g_ab',  'metric.export')         # cdb (metric,tmp)

tmp = cdblib.get ('g_ab_scaled0','metric.export')   # cdb (metric.scaled0,tmp)
tmp = cdblib.get ('g_ab_scaled2','metric.export')   # cdb (metric.scaled2,tmp)
tmp = cdblib.get ('g_ab_scaled3','metric.export')   # cdb (metric.scaled3,tmp)
tmp = cdblib.get ('g_ab_scaled4','metric.export')   # cdb (metric.scaled4,tmp)
tmp = cdblib.get ('g_ab_scaled5','metric.export')   # cdb (metric.scaled5,tmp)

# ------------------------------------------------------------------------------
# the inverse metric

# inverse metric in terms of R and \partial R
tmp = cdblib.get ('g^ab_3','metric-inv.export')         # cdb (metric3.inv,tmp)
tmp = cdblib.get ('g^ab_4','metric-inv.export')         # cdb (metric4.inv,tmp)
tmp = cdblib.get ('g^ab_5','metric-inv.export')         # cdb (metric5.inv,tmp)
tmp = cdblib.get ('g^ab_6','metric-inv.export')         # cdb (metric6.inv,tmp)
```

```
# inverse metric in terms of R and \nabla R
tmp = cdblib.get ('g^ab',  'metric-inv.export')         # cdb (metric.inv,tmp)

tmp = cdblib.get ('g^ab_scaled0','metric-inv.export')   # cdb (metric.inv.scaled0,tmp)
tmp = cdblib.get ('g^ab_scaled2','metric-inv.export')   # cdb (metric.inv.scaled2,tmp)
tmp = cdblib.get ('g^ab_scaled3','metric-inv.export')   # cdb (metric.inv.scaled3,tmp)
tmp = cdblib.get ('g^ab_scaled4','metric-inv.export')   # cdb (metric.inv.scaled4,tmp)
tmp = cdblib.get ('g^ab_scaled5','metric-inv.export')   # cdb (metric.inv.scaled5,tmp)

# ------------------------------------------------------------------------------
# the generalised connections

# 4th order gen gamma
tmp = cdblib.get ('gen_gamma_0_4th','genGamma.export')  # cdb (genGamma04,tmp)
tmp = cdblib.get ('gen_gamma_1_4th','genGamma.export')  # cdb (genGamma14,tmp)

# 6th order gen gamma
tmp = cdblib.get ('gen_gamma_0','genGamma.export')  # cdb (genGamma0,tmp)
tmp = cdblib.get ('gen_gamma_1','genGamma.export')  # cdb (genGamma1,tmp)
tmp = cdblib.get ('gen_gamma_2','genGamma.export')  # cdb (genGamma2,tmp)
tmp = cdblib.get ('gen_gamma_3','genGamma.export')  # cdb (genGamma3,tmp)

# 6th order gen gamma scaled
tmp = cdblib.get ('gen_gamma_0_scaled','genGamma.export')  # cdb (genGamma0scaled,tmp)
tmp = cdblib.get ('gen_gamma_1_scaled','genGamma.export')  # cdb (genGamma1scaled,tmp)
tmp = cdblib.get ('gen_gamma_2_scaled','genGamma.export')  # cdb (genGamma2scaled,tmp)
tmp = cdblib.get ('gen_gamma_3_scaled','genGamma.export')  # cdb (genGamma3scaled,tmp)

# gen gamma in terms of partial derivs of Gamma^{a}_{bc}
tmp = cdblib.get ('gen_gamma_pderiv0','genGamma.export')  # cdb (genGammaPderiv0,tmp)
tmp = cdblib.get ('gen_gamma_pderiv1','genGamma.export')  # cdb (genGammaPderiv1,tmp)
tmp = cdblib.get ('gen_gamma_pderiv2','genGamma.export')  # cdb (genGammaPderiv2,tmp)

# ------------------------------------------------------------------------------
# the metric determinent and friends

tmp = cdblib.get ('Ndetg','detg2.export')      # cdb (Ndetg,tmp)
tmp = cdblib.get ('sqrtNdetg','detg2.export')  # cdb (sqrtNdetg,tmp)
```

```
tmp = cdblib.get ('logNdetg','detg2.export')   # cdb (logNdetg,tmp)

# ------------------------------------------------------------------------------
# the geodesic ivp

# 4th order ivp terms, scaled
tmp = cdblib.get ('ivp42','geodesic-ivp.export')  # cdb (ivp42,tmp)
tmp = cdblib.get ('ivp43','geodesic-ivp.export')  # cdb (ivp43,tmp)

# 6th order ivp terms, scaled
tmp = cdblib.get ('ivp62','geodesic-ivp.export')  # cdb (ivp62,tmp)
tmp = cdblib.get ('ivp63','geodesic-ivp.export')  # cdb (ivp63,tmp)
tmp = cdblib.get ('ivp64','geodesic-ivp.export')  # cdb (ivp64,tmp)
tmp = cdblib.get ('ivp65','geodesic-ivp.export')  # cdb (ivp65,tmp)

# ------------------------------------------------------------------------------
# the geodesic bvp

# 4th order ivp
tmp = cdblib.get ('bvp4','geodesic-bvp.export')  # cdb (bvp4,tmp)

# 6th order bvp terms, scaled
tmp = cdblib.get ('bvp622','geodesic-bvp.export')  # cdb (bvp622,tmp)
tmp = cdblib.get ('bvp623','geodesic-bvp.export')  # cdb (bvp623,tmp)
tmp = cdblib.get ('bvp624','geodesic-bvp.export')  # cdb (bvp624,tmp)
tmp = cdblib.get ('bvp625','geodesic-bvp.export')  # cdb (bvp625,tmp)

tmp = cdblib.get ('bvp633','geodesic-bvp.export')  # cdb (bvp633,tmp)
tmp = cdblib.get ('bvp634','geodesic-bvp.export')  # cdb (bvp634,tmp)
tmp = cdblib.get ('bvp635','geodesic-bvp.export')  # cdb (bvp635,tmp)

tmp = cdblib.get ('bvp644','geodesic-bvp.export')  # cdb (bvp644,tmp)
tmp = cdblib.get ('bvp645','geodesic-bvp.export')  # cdb (bvp645,tmp)

tmp = cdblib.get ('bvp655','geodesic-bvp.export')  # cdb (bvp655,tmp)

# ------------------------------------------------------------------------------
# the geodesic lsq
```

```
# 6th order lsq terms, scaled
tmp = cdblib.get ('rnc61scaled','gen2rnc.export')  # cdb (rnc61,tmp)
tmp = cdblib.get ('rnc62scaled','gen2rnc.export')  # cdb (rnc62,tmp)
tmp = cdblib.get ('rnc63scaled','gen2rnc.export')  # cdb (rnc63,tmp)
tmp = cdblib.get ('rnc64scaled','gen2rnc.export')  # cdb (rnc64,tmp)
tmp = cdblib.get ('rnc65scaled','gen2rnc.export')  # cdb (rnc65,tmp)

# ------------------------------------------------------------------------------
# the geodesic lsq

# 3rd to 6th order lsq
tmp = cdblib.get ('lsq3','geodesic-lsq.export')  # cdb (lsq3,tmp)
tmp = cdblib.get ('lsq4','geodesic-lsq.export')  # cdb (lsq4,tmp)
tmp = cdblib.get ('lsq5','geodesic-lsq.export')  # cdb (lsq5,tmp)
tmp = cdblib.get ('lsq6','geodesic-lsq.export')  # cdb (lsq6,tmp)

# 6th order lsq terms, scaled
tmp = cdblib.get ('lsq60','geodesic-lsq.export')  # cdb (lsq60,tmp)
tmp = cdblib.get ('lsq62','geodesic-lsq.export')  # cdb (lsq62,tmp)
tmp = cdblib.get ('lsq63','geodesic-lsq.export')  # cdb (lsq63,tmp)
tmp = cdblib.get ('lsq64','geodesic-lsq.export')  # cdb (lsq64,tmp)
tmp = cdblib.get ('lsq65','geodesic-lsq.export')  # cdb (lsq65,tmp)

# ------------------------------------------------------------------------------
# the dRabcd

# 6th order dRabcd, scaled
tmp = cdblib.get ('dRabcd61scaled','dRabcd.export')  # cdb (dRabcd61,tmp)
tmp = cdblib.get ('dRabcd62scaled','dRabcd.export')  # cdb (dRabcd62,tmp)
tmp = cdblib.get ('dRabcd63scaled','dRabcd.export')  # cdb (dRabcd63,tmp)
tmp = cdblib.get ('dRabcd64scaled','dRabcd.export')  # cdb (dRabcd64,tmp)
tmp = cdblib.get ('dRabcd65scaled','dRabcd.export')  # cdb (dRabcd65,tmp)

# ------------------------------------------------------------------------------
# the dGamma

# 6th order dGamma, scaled
```

```
tmp = cdblib.get ('dGamma61scaled','dGamma.export')  # cdb (dGamma61,tmp)
tmp = cdblib.get ('dGamma62scaled','dGamma.export')  # cdb (dGamma62,tmp)
tmp = cdblib.get ('dGamma63scaled','dGamma.export')  # cdb (dGamma63,tmp)
tmp = cdblib.get ('dGamma64scaled','dGamma.export')  # cdb (dGamma64,tmp)
tmp = cdblib.get ('dGamma65scaled','dGamma.export')  # cdb (dGamma65,tmp)
```