```
vc_mng_wf  THM

⊢  ∀v:vc{i:l}(), ds,da:Collection(dec()), de:sig(), rho:Decl
|  , e:{[[de]] rho}, s:{[[ds]] rho}, tr:trace_env([[da]] rho).
|  tc_vc(v;ds;da;de)
|  ⇒
|  trace_consistent_vc(rho;da;tr.proj;v)
|  ⇒
|  vc_mng(v;rho;ds;da;de;e;s;tr) ∈ Prop
| by Unfold `trace_consistent_vc` 0 THEN

|    Unfolds ``vc_hyp vc_concl vc_mng`` 0 THEN  Auto

|    THEN Try (Unfold `decl` 0 THEN Auto)

|    THEN D 1 THEN AllHyps (Unfold `tc_vc`) THEN All Reduce

|    THEN Try (Complete Auto)
|
•
|
|\
| 1. x : imp{i:l}()
| 2. ds : Collection(dec())
| 3. da : Collection(dec())
| 4. de : sig()
| 5. rho : Decl
| 6. e : {[[de]] rho}
| 7. s : {[[ds]] rho}
| 8. tr : trace_env([[da]] rho)
| 9. tc_pred(x.hyp;ds;<>;de) & tc_pred(x.concl;ds;<>;de)
| 10. trace_consistent_pred(rho;da;tr.proj;x.hyp)
| 11. trace_consistent_pred(rho;da;tr.proj;x.concl)
| ⊢  ([[x.hyp]] rho ds <> de e s · mk_trace_env(nil, tr.proj)
|     ⇒
|     [[x.concl]] rho ds <> de e s · mk_trace_env(nil, tr.proj))
|    ∈ Prop
|   by MemCD THEN Using [`daa`,da
|      ] (BackThruLemma `pred_mng_wf` THEN Auto)
|
|      THEN Try (BackThruLemma `empty_sts_mng` THEN Auto)
|
|      THEN Reduce 0 THEN Auto•
|
.\
. 1. y : qimp{i:l}()
. 2. ds : Collection(dec())
. 3. da : Collection(dec())
. 4. de : sig()
. 5. rho : Decl
. 6. e : {[[de]] rho}
. 7. s : {[[ds]] rho}
. 8. tr : trace_env([[da]] rho)
. 9. tc_pred(y.hyp;ds;dec_lookup(da;y.lbl);de)
. |  & tc_pred(y.concl;ds;dec_lookup(da;y.lbl);de)
```

```
. 10. trace_consistent_pred(rho;da;tr.proj;y.hyp)
. 11. trace_consistent_pred(rho;da;tr.proj;y.concl)
. ⊢  ...
. |  ∈ Prop
. | by MemCD THENL [Auto; MemCD]
.
. |    THEN BackThruLemma ‘pred_mng_wf‘ THEN Auto
.
. |    THEN Try (Unfold ‘tappend‘ 0) THEN Reduce 0
. THEN Try Trivial
.
. |    ⋅
. |
. .....subterm .....
. 1. y : qimp{i:l}()
. 2. ds : Collection(dec())
. 3. da : Collection(dec())
. 4. de : sig()
. 5. rho : Decl
. 6. e : {[[de]] rho}
. 7. s : {[[ds]] rho}
. 8. tr : trace_env([[da]] rho)
. 9. tc_pred(y.hyp;ds;dec_lookup(da;y.lbl);de)
. 10. tc_pred(y.concl;ds;dec_lookup(da;y.lbl);de)
. 11. trace_consistent_pred(rho;da;tr.proj;y.hyp)
. 12. trace_consistent_pred(rho;da;tr.proj;y.concl)
. 13. v : [[dec_lookup(da;y.lbl)]] rho
. 14. [[y.hyp]] rho ds dec_lookup(da;y.lbl) de e s v tr
. ⊢  <y.lbl,v> ∈ (Σ[[da]] rho)
.   by BackThruLemma’ ‘sigma_decls_mng2‘ THEN Auto
END
```