

File created: 15-Aug-90 17:54:09 {DSK}<usr>local>lde>SOURCES>loops>TRUCKIN>LOOPSGATEWAY.;2

changes to: (VARS LOOPSGATEWAYCOMS)
(FNS PM.Expunge PM.AddToDL PM.DelFromDLname PM.DelFromDL PM.AddToMailQ PM.DistributeMail)

previous date: 17-Dec-87 08:29:53 {DSK}<usr>local>lde>SOURCES>loops>TRUCKIN>LOOPSGATEWAY.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1983, 1985, 1987, 1990 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **LOOPSGATEWAYCOMS**

```
[(DECLARE%: DONTCOPY (PROP MAKEFILE-ENVIRONMENT LOOPSGATEWAY))
 (CLASSES Gateway MasterGateway SlaveGateway)
 (COMS (* Items necessary for Postal Users)
  (INITVARS (\MY.NSHOSTID NIL)
   (\My.PupAddress NIL)
   (\PostMaster.PupAddress NIL)
   (POSTMAN NIL))
  (GLOBALVARS \MY.NSHOSTNUMBER \MY.NSHOSTID \My.PupAddress \PostMaster.PupAddress POSTMAN)
  (DECLARE%: DONTCOPY (MACROS PostMasterPupAddress HostAddresses))
  (FNS RPCToPO ReceiveMail ReceiveMailLst))
 (COMS (* Items necessary for PostMaster)
  (INITVARS (ALL.DL'S NIL)
   (\MY.NSHOSTID NIL)
   (\My.PupAddress NIL)
   (PM.CATATONIC.WAIT.ms 10000))
  (GLOBALVARS \MY.NSHOSTNUMBER ALL.DL'S \MY.NSHOSTID \My.PupAddress PM.CATATONIC.WAIT.ms)
  (DECLARE%: DONTCOPY (MACROS HostAddresses))
  (RECORDS PostalDL DLEntry)
  (FNS PM.EstablishServices PM.Expunge PM.AddToDL PM.DelFromDLname PM.DelFromDL PM.AddToMailQ
   RPCToGateway POSTMONITOR)
  (INITVARS (PM.StatsTimeDelta.ms 10000)
   (%#MsgsDelivered 0)
   (%#MsgsInSingleBag 0)
   (%#MsgsInSingleBag 0.0)
   (DeliveryAttempts 0)
   (DeliveryTimeouts 0)
   (%#DeliveryTimeout 0.0)
   (AvMultiBagLen 0)
   (DeliveryHistory NIL))
  (GLOBALVARS PM.StatsTimeDelta.ms %#MsgsDelivered %#MsgsInSingleBag %#MsgsInSingleBag
   DeliveryAttempts DeliveryTimeouts %#DeliveryTimeout AvMultiBagLen NIL)
  (FNS PM.InitDL PM.InitPostalServices PM.DistributeMail))
 (METHODS Gateway.AddMeToDL Gateway.GetPostalBox Gateway.Out Gateway.ProcessOutQ
  Gateway.Quit Gateway.Receive Gateway.ReceiveP Gateway.Send Gateway.SendP Gateway.StoreIn
  MasterGateway.GetPostalBox SlaveGateway.GetPostalBox)
 (DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS (ADDVARS (NLAMA PM.EstablishServices)
  (NLAML PM.DistributeMail
   PM.AddToMailQ
   PM.DelFromDLname PM.AddToDL)
  (LAMA]))
```

(DECLARE%: DONTCOPY

(PUTPROPS **LOOPSGATEWAY MAKEFILE-ENVIRONMENT** (:PACKAGE "IL" :READTABLE "INTERLISP" :BASE 10))
)

(DEFCLASSES Gateway MasterGateway SlaveGateway)

```
(DEFCLASS Gateway (MetaClass Class Edited%: (* dgb%: "12-JUL-83 16:56")
  doc "This is the default metaClass for all classes")
  (Supers Object)
  (InstanceVariables (inQ NIL doc "queue of messages coming in from outside. lastItem property is pointer
    to lastItem on queue" lastItem NIL)
   (outQ NIL doc "queue of messages to be sent out. lastItem property is pointer to lastItem on
    queue" lastItem NIL)
   (outProcess NIL doc "Process serving the outgoing queue")
   (postalName NIL doc "name to appear on PostBox")
   (postMasterAddresses NIL doc "List of pupHostnum and NSHostID for correspondent site.")
   (postIDlst NIL doc "cached list used as args for REMOTEVAL")
   (receiver NIL doc "name of postMaster distribution list which will be receiving mail from you")
   (lastSeqNum 0 doc "Sequence number of the last message into inQ")
   (lastOutNum 0 doc "Sequence number of last message sent from outQ")))
```

```
(DEFCLASS MasterGateway (MetaClass Class Edited%: (* dgb%: "5-JUL-83 17:14")
  doc "This is the default metaClass for all classes")
  (Supers Gateway))
```

```
(DEFCLASS SlaveGateway (MetaClass Class Edited%: (* dgb%: "6-JUL-83 13:01")
  doc "This is the default metaClass for all classes")
```

(Supers Gateway))

(* * Items necessary for Postal Users)

(RPAQ? \MY.NSHOSTID NIL)

(RPAQ? \My.PupAddress NIL)

(RPAQ? \PostMaster.PupAddress NIL)

(RPAQ? POSTMAN NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \MY.NSHOSTNUMBER \MY.NSHOSTID \My.PupAddress \PostMaster.PupAddress POSTMAN)
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS PostMasterPupAddress MACRO (NIL (CAR (@ postMasterAddresses))))

(PUTPROPS HostAddresses MACRO (NIL (LIST (ETHERHOSTNUMBER)
 \MY.NSHOSTID)))
)

(DEFINEQ

(RPCtoPO

[LAMBDA (Fun ArgList multiplicity)
 (COND ((EQ \My.PupAddress \PostMaster.PupAddress)
 (APPLY Fun ArgList))

(T (REMOTEAPPLY Fun ArgList \PostMaster.PupAddress multiplicity]))

(* dgb%: "7-JUL-83 20:46")

(ReceiveMail

[LAMBDA (seqNum MSG)

(* edited%: "14-JUL-83 02:04")

(* Stores a message coming in over the net in a queue for later
servicing)

(_ POSTMAN StoreIn seqNum MSG])

(ReceiveMailLst

[LAMBDA (seqNum MSGLST)

(* edited%: "14-JUL-83 02:10")

(* Stores a list of messages coming in over the net in a queue for later servicing)

 (_ POSTMAN StoreIn seqNum MSGLST T])
)

(* * Items necessary for PostMaster)

(RPAQ? ALL.DL'S NIL)

(RPAQ? \MY.NSHOSTID NIL)

(RPAQ? \My.PupAddress NIL)

(RPAQ? PM.CATATONIC.WAIT.ms 10000)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \MY.NSHOSTNUMBER ALL.DL'S \MY.NSHOSTID \My.PupAddress PM.CATATONIC.WAIT.ms)
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS HostAddresses MACRO (NIL (LIST (ETHERHOSTNUMBER)
 \MY.NSHOSTID)))
)

(DECLARE%: EVAL@COMPILE

(DATATYPE PostalDL ((DLName POINTER)
 (SendToFilterFN POINTER))(* The SendToFilterFN field is a function of two args -- DL name and POSTALNAME of sender --
which must return non-NIL if the "mailing" is permitted.)

```

        (UpdateFilterFN POINTER)

(* The UpdateFilterFN is a function of three args -- POSTALNAME of sender, DL name, and reason
(either ADD or DELETE) -- which must return non-NIL if the updating is permitted.)

        (MSGQ POINTER)

(* The MSGQ field is a TCONC queue, where the first item is merely NIL --
The Patrons field is a list of DLEntry items)

        (Patrons POINTER)
        (LetterWritersAlist POINTER)
        (PostEVENT POINTER))
SendToFilterFN _ (FUNCTION TRUE)
UpdateFilterFN _ (FUNCTION TRUE)
MSGQ _ (TCONC (LIST NIL))
PostEVENT _ (CREATE.EVENT "ReceivedMail"))

(RECORD DLEntry (POSTALNAME lastSeqNum MSGQptr PUPNUM NSHOSTID DLProc)

(* The MSGQptr points into tails of the Q (in TCONC form) of messages sent to this DL.
CAR of the ptr is the most recent message sent to this patron (first message is NIL))

)

(/DECLAREDATATYPE 'PostalDL ' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
'14)

(DEFINEQ

PM.EstablishServices
[NLAMBDA L
(* dgb%: " 6-JUL-83 09:24")
(* Machines with boxes are those on the DL PostalPatrons)

(PROG [(POSTALNAME (APPLY 'PM.AddToDL (CONS 'PostalPatrons L)
(COND
(POSTALNAME (RETURN (CONS POSTALNAME (HostAddresses))

PM.Expunge
[LAMBDA (NAME)
; Edited 15-Aug-90 17:48 by jds
[for DLNAME PAIR DL in ALL.DL'S do (PM.DelFromDL NAME (SETQ DL (GETTOPVAL DLNAME)))
(AND (SETQ PAIR (FASSOC NAME (fetch (PostalDL LetterWritersAlist)
of DL)))
(replace (PostalDL LetterWritersAlist) of DL
with (DREMOVE PAIR (fetch (PostalDL LetterWritersAlist) of DL)
T])

PM.AddToDL
[NLAMBDA (DLNAME NAME PUPNUM NSHOSTID)
; Edited 15-Aug-90 17:49 by jds

(* If successful, returns the Postal NAME which was added; returns NIL if the UpdateFilterFN rejects the request.
Errors out if DLNAME isn't a postal DL, or if there is already an entry with the same Postal Name but for a different address.)

(PROG ((DL (GETTOPVAL DLNAME))
ENTRY PROCNAME)
(if (NOT (type? PostalDL DL))
then (ERROR "Not a DL name " DLNAME)
elseif (NULL (APPLY* (fetch (PostalDL UpdateFilterFN) of DL)
NAME DLNAME 'ADD))
then (RETURN))
[if (SETQ ENTRY (ASSOC NAME (fetch (PostalDL Patrons) of DL)))
then (if [NOT (OR (EQ PUPNUM (fetch (DLEntry PUPNUM) of ENTRY))
(EQ NSHOSTID (fetch (DLEntry NSHOSTID) of ENTRY))
then (ERROR "Machine Name already in use"))
else (push (fetch (PostalDL Patrons) of DL)
(SETQ ENTRY (create DLEntry]
(create DLEntry smashing ENTRY POSTALNAME _ NAME PUPNUM _ PUPNUM NSHOSTID _ NSHOSTID MSGQptr _
(CAR (fetch (PostalDL MSGQ) of DL))
lastSeqNum _ 0)
(if (NEQ DLNAME 'PostalPatrons)
then
(* Currently, there is no mailing allowed to all Postal patrons.)
(SETQ PROCNAME (PACK* NAME '%. DLNAME '.Mailer))
(DEL.PROCESS (FIND.PROCESS PROCNAME))
(SETQ PROCNAME (ADD.PROCESS (LIST 'PM.DistributeMail NAME DLNAME)
'NAME PROCNAME 'RESTARTABLE T))
(replace (DLEntry DLProc) of ENTRY with PROCNAME)
(NOTIFY.EVENT (fetch (PostalDL PostEVENT) of DL)))
(RETURN NAME)])

PM.DelFromDLname
[NLAMBDA (DLNAME NAME PUPADDR NSHOSTID)
; Edited 15-Aug-90 17:48 by jds

```

(* If successful, returns the Postal NAME which was deleted; returns NIL if the UpdateFilterFN rejects the request.
Errors out if DLNAME isn't a postal DL, or if there is already an entry with the same Postal Name but for a different address.)

```
(PROG ((DL (GETTOPVAL DLNAME)))
  (COND
    ((NOT (type? PostalDL DL))
      (ERROR "Not a DL name " DLNAME))
    ((NULL (APPLY* (fetch (PostalDL UpdateFilterFN) of DL)
      NAME DLNAME 'DELETE)))
    (RETURN)))
  (PM.Del.aux NAME DL PUPADDR NSHOSTID)
  (RETURN NAME])
```

(PM.DelFromDL

```
[LAMBDA (NAME DL PUPADDR NSHOSTID) ; Edited 15-Aug-90 17:49 by jds
  (PROG [(ENTRY (ASSOC NAME (fetch (PostalDL Patrons) of DL)
    (COND
      (ENTRY (AND (OR PUPADDR NSHOSTID)
        (NOT (EQ PUPADDR (fetch (DLEntry PUPNUM) of ENTRY)))
        (NOT (EQ NSHOSTID (fetch (DLEntry NSHOSTID) of ENTRY)))
        (ERROR "Machine Name conflict when deleteing from DL"))
      )
      (DEL.PROCESS (fetch (DLEntry DLProc) of ENTRY))
      (replace (PostalDL Patrons) of DL with (DREMOVE ENTRY (fetch (PostalDL Patrons) of DL))
```

(* At least one of the specified address is non-null, but neither one matches this DLEntry)

```
)
  (DEL.PROCESS (fetch (DLEntry DLProc) of ENTRY))
  (replace (PostalDL Patrons) of DL with (DREMOVE ENTRY (fetch (PostalDL Patrons) of DL))
```

(PM.AddToMailQ

```
[NLAMBDA (DLNAME outNum MSG POSTALNAME) ; Edited 15-Aug-90 17:48 by jds
```

(* adds MSG to Queue on distribution list. Each patron knows how much they have seen of the Q)

```
(PROG (WriterNumPair (DL (GETTOPVAL DLNAME)))
  (COND
    ((NOT (type? PostalDL DL))
      (ERROR "Not a DL name " DLNAME))
    ((NULL (APPLY* (fetch (PostalDL SendToFilterFN) of DL)
      DLNAME POSTALNAME))
      (RETURN)))
    (COND
      ([NULL (SETQ WriterNumPair (FASSOC POSTALNAME (fetch (PostalDL LetterWritersAlist) of DL)
        (SETQ WriterNumPair (CONS POSTALNAME 0))
        (push (fetch (PostalDL LetterWritersAlist) of DL)
          WriterNumPair)))
        (COND
          ((IEQP outNum (ADD1 (CDR WriterNumPair)))
            (TCONC (fetch (PostalDL MSGQ) of DL)
              MSG)
            (add (CDR WriterNumPair)
              1)
            (NOTIFY.EVENT (fetch (PostalDL PostEVENT) of DL)))
          ((NOT (IEQP outNum (CDR WriterNumPair)))
            (HELP outNum "Out of sequence")))
          (RETURN T])
```

(RPCtoGateway

```
[LAMBDA (Fun ArgList Gateway.PupAddress multiplicity) (* dgb%: " 7-JUL-83 20:15")
  (COND
    ((EQ \My.PupAddress Gateway.PupAddress)
      (APPLY Fun ArgList))
    (T (REMOTEAPPLY Fun ArgList Gateway.PupAddress multiplicity])
```

(POSTMONITOR

```
[LAMBDA NIL (* JonL " 8-AUG-83 20:26")
  (PROG NIL
    LOOP
      (BLOCK PM.StatsTimeDelta.ms)
      [SETQ AvMultiBagLen (FPLUS (FTIMES 0.625 AvMultiBagLen)
        (FTIMES 0.375 (FQUOTIENT (IDIFFERENCE %MsgsDelivered %MsgsInSingleBag)
          (IDIFFERENCE DeliveryAttempts DeliveryTimeouts))
        (SETQ %MsgsInSingleBag (FPLUS (FTIMES 0.625 %MsgsInSingleBag)
          (FTIMES 0.375 (FQUOTIENT %MsgsInSingleBag %MsgsDelivered))
          (SETQ %MsgsDelivered 0)
          (SETQ %MsgsInSingleBag 0)
          [SETQ %DeliveryTimeout (FPLUS (FTIMES 0.625 %DeliveryTimeout)
            (FTIMES 0.375 (FQUOTIENT DeliveryTimeouts DeliveryAttempts))
            (push DeliveryHistory (LIST AvMultiBagLen %MsgsInSingleBag %DeliveryTimeout))
            (SETQ DeliveryAttempts 0)
            (SETQ DeliveryTimeouts 0)
            (GO LOOP))
```

)

(RPAQ? **PM.StatsTimeDelta.ms** 10000)(RPAQ? **%#MsgsDelivered** 0)(RPAQ? **%#MsgsInSingleBag** 0)(RPAQ? **%%MsgsInSingleBag** 0.0)(RPAQ? **DeliveryAttempts** 0)(RPAQ? **DeliveryTimeouts** 0)(RPAQ? **%%DeliveryTimeout** 0.0)(RPAQ? **AvMultiBagLen** 0)(RPAQ? **DeliveryHistory** NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

```
(GLOBALVARS PM.StatsTimeDelta.ms %#MsgsDelivered %#MsgsInSingleBag %%MsgsInSingleBag DeliveryAttempts
  DeliveryTimeouts %%DeliveryTimeout AvMultiBagLen NIL)
)
```

(DEFINEQ

(PM.InitDL

[LAMBDA (DLNAME SendToFilterFN UpdateFilterFN)

(* dgb%: " 6-JUL-83 21:18")

(* Adds DL to the list of ALL.DL'S. Error if previously existing. Creates a process which will service each of the patrons on this mail queue. Deletes previously existing processes of the same name)

(AND (FMEMB DLNAME ALL.DL'S)

(ERROR "DL name already in use" DLNAME))

[SETTOPVAL DLNAME (**create** PostalDL

DLName _ DLNAME

SendToFilterFN _ (OR SendToFilterFN (FUNCTION TRUE))

UpdateFilterFN _ (OR UpdateFilterFN (FUNCTION TRUE)]

(**push** ALL.DL'S DLNAME)

DLNAME])

(PM.InitPostalServices

[LAMBDA NIL

(* JonL " 8-AUG-83 20:23")

(* Clears ALL.DL'S and sets up the initial distribution lists)

(SETQ EvalServerGaggedHosts)

(* Just in case we want to REMOTEVAL to ourselves when initially establishing Gateway postal services)

[OR (AND \MY.NSHOSTID (LITATOM \MY.NSHOSTID))

(SETQ \MY.NSHOSTID (CONCAT "#NSID." (CADR \MY.NSHOSTNUMBER)

".")

(CADDR \MY.NSHOSTNUMBER)

".")

(CADDRR \MY.NSHOSTNUMBER)]

(OR (SMALLP \My.PupAddress)

(SETQ \My.PupAddress (ETHERHOSTNUMBER)))

[MAP.PROCESSES (FUNCTION (LAMBDA (HANDLE NAME)

(COND

((EQ (STRPOS '.Mailer NAME NIL NIL NIL T)

(ADD1 (NCHARS NAME)))

(DEL.PROCESS HANDLE]

(OR (FIND.PROCESS 'EvalServerListening)

(EVALSERVER))

(DEL.PROCESS (FIND.PROCESS 'POSTMONITOR))

(SETQ %#MsgsDelivered 0)

(SETQ %#MsgsInSingleBag 0)

(SETQ %%MsgsInSingleBag 0.0)

(SETQ DeliveryAttempts 0)

(SETQ DeliveryTimeouts 0)

(SETQ %%DeliveryTimeout 0.0)

(SETQ DeliveryHistory)

(ADD.PROCESS ' (**POSTMONITOR**)

'RESTARTABLE T)

(SETTOPVAL 'ALL.DL'S)

(**PM.InitDL** 'PostalPatrons)(**PM.InitDL** 'WorldMaster)(**PM.InitDL** 'Players)

(LIST \My.PupAddress \MY.NSHOSTID])

(PM.DistributeMail

[NLAMBDA (PatronNAME DLNAME)

; Edited 15-Aug-90 17:49 by jds

(* Send out one message for one patron on this distribution list and then block.
If no mail wait for event)

```

(PROG (PatronDLEntry nextMsgPtr nextSeqNum msgLst (DL (GETTOPVAL DLNAME)))
  (SETQ PatronDLEntry (ASSOC PatronNAME (fetch (PostalDL Patrons) of DL)))
  1PASS
  [COND
    [msgLst
      (* AHA! We've already bagged somethin up)
      (SETQ nextSeqNum (ADD1 (fetch (DLEntry lastSeqNum) of PatronDLEntry)))
      (BLOCK (COND
        ((PROG1 (NLSETQ (RPCtoGateway 'ReceiveMailLst (LIST nextSeqNum msgLst)
          (fetch (DLEntry PUPNUM) of PatronDLEntry)
          0))
          (add DeliveryAttempts 1))
        (* Delivering the mail means sending it, and updating the patrons MSGQ pointer to the next message)
          (replace (DLEntry MSGQptr) of PatronDLEntry with nextMsgPtr)
          (replace (DLEntry lastSeqNum) of PatronDLEntry with nextSeqNum)
          (PROG ((LEN (LENGTH msgLst)))
            (add %MsgsDelivered LEN)
            (if (EQ 1 LEN)
              then (add %MsgsInSingleBag 1)))
          (SETQ msgLst))
          (T (add DeliveryTimeouts 1)

(* If the RPC fails to get thru, then perhaps the client is "catatonic" and we should wait a while.)
          PM.CATATONIC.WAIT.ms]
        ([NULL (SETQ nextMsgPtr (CDR (fetch (DLEntry MSGQptr) of PatronDLEntry)
          (AWAIT.EVENT (fetch (PostalDL PostEVENT) of DL)
            5000)
            (* Notice that we check every five seconds or so just to be sure.)
          )
          (* Try to bag up as many msgs as possible into the msgLst)
          (T
            (PROG ([lastCell (SETQ msgLst (LIST (CAR nextMsgPtr]
              (lenRemaining 512)
              (thisLen NIL))
              LP (COND
                ([OR (NULL (CDR nextMsgPtr))
                  (ILEQ [SETQ lenRemaining (IDIFFERENCE lenRemaining
                    (IPLUS 2 (OR thisLen (NCHARS (CAR nextMsgPtr]
                    (SETQ thisLen (NCHARS (CADR nextMsgPtr]
                    (* Note how cleverly this code omits calling NCHARS when there is only one item in the msgQ)
                    (RETURN)))
                    (pop nextMsgPtr)
                    [SETQ lastCell (CDR (RPLACD lastCell (LIST (CAR nextMsgPtr]
                    (GO LP]
                    (GO 1PASS])
                    )
                    (\BatchMethodDefs)
                    (\UnbatchMethodDefs)
                    (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS
                    (ADDTOVAR NLAMA PM.EstablishServices)
                    (ADDTOVAR NLAML PM.DistributeMail PM.AddToMailQ PM.DelFromDLname PM.AddToDL)
                    (ADDTOVAR LAMA )
                    )
                    (PUTPROPS LOOPSGATEWAY COPYRIGHT ("Venue & Xerox Corporation" 1983 1985 1987 1990))

```

FUNCTION INDEX

PM.AddToDL	3	PM.DistributeMail	5	PM.InitPostalServices ...	5	RPCtoGateway	4
PM.AddToMailQ	4	PM.EstablishServices	3	POSTMONITOR	4	RPCtoPO	2
PM.DelFromDL	4	PM.Expunge	3	ReceiveMail	2		
PM.DelFromDLname	3	PM.InitDL	5	ReceiveMailLst	2		

VARIABLE INDEX

%#MsgsDelivered	5	ALL.DL'S	2	DeliveryTimeouts	5	\MY.NSHOSTID	2
%#MsgsInSingleBag	5	AvMultiBagLen	5	PM.CATATONIC.WAIT.ms	2	\My.PupAddress	2
%%DeliveryTimeout	5	DeliveryAttempts	5	PM.StatsTimeDelta.ms	5	\PostMaster.PupAddress ..	2
%%MsgsInSingleBag	5	DeliveryHistory	5	POSTMAN	2		

RECORD INDEX

DLEntry	3	PostalDL	2
---------------	---	----------------	---

MACRO INDEX

HostAddresses	2	PostMasterPupAddress	2
---------------------	---	---------------------------	---

PROPERTY INDEX

LOOPSGATEWAY	1
--------------------	---