```
!
! Copyright (c) 2008 Sage Software Canada Ltd. All Rights Reserved
! Style Checker - generic style check

! ** This class is created by an application developer who wishes to enable the
! ** SytleCheck capability that is build into the ProvideX plug-in.

! ** @note The primary purpose of this class is to load the developer-specific
! ** @note rules and to provide additional logic to handle special cases that
! ** @note cannot be described using the standard rules.
def class "genStyleCheck"  create required
    like "ecStyleCheck"

    ! specify the full path to the location of the file that contains the standard
    ! style rules this is usually in the same folder as the class file
    local theStyleRules$=mid(pgn,1,pos(dlm=pgn,-1))+"genStyleCheck_rules.txt"

    ! ** This method will be called when the 'Configure' action is triggered from
    ! ** the contributed tools menu.
    ! **
    ! ** @param inActionEventID A reference to %JavaClass(eventmanager/ExtCommand)%
    function configPerformed(inActionEventID)                do_configperformed

    ! ** Get a description for this listener that will be used if this class doubles
    ! ** as an observer.  This will be determined by the EventNotificationFlag.
    ! ** @returns A string containing the description of this listener
    function getDescription$()                               get_description

    ! ** Check the program line against the defined rules and return the modified
    ! ** line and failed rule
    ! ** @param line$ The program line to be checked; will return the modified line
    ! ** @param searchVal$ The search value associated with the rule that failed
    ! ** @param severity$ The severity that is associated with the rule
    ! ** @param desc$ Used to return the description of the rule that failed
    ! ** @returns A non-zero value to indicate that a rule failed.
    ! ** @value heading=Severity Codes
    ! ** @value E Error
    ! ** @value W Warning
    function local CheckStyle(line$,searchVal$,severity$,desc$) check_style

    ! ** The logic to be executed when the observer is triggered during the normal
    ! ** event process in the ProvideX event manager.  This logic must check the
    ! ** major and minor codes to determine the current event and then decide what
    ! ** action is to be performed.
    ! **
    ! ** @param state A reference to an object of class %PvxClass(PvxState)%
    function update( state )                                 do_update

end def

on_create:
    ! Use the method from the plug-in class to load the rules
    _obj'loadRules(from "ecStyleCheck", theStyleRules$)
return

get_description:
    ! Set the description for this observer
    local theDescription$="Generic Style Check"
return theDescription$

check_style:
enter line$,searchVal$,severity$,desc$
```

```
    local stdStyleCheck
    retVAL = 0

    ! Save status returned by the CheckStyle() method
    retVAL=_obj'CheckStyle(from "ecStyleCheck", line$, searchVal$, severity$, desc$)

    ! Add specific logic for additional checks to be performed for each program line

return retVAL

! The configuration option was selected from the contributed tools menu; this routine
! executed the method from the plug-in class
do_configperformed:
enter inActionEventID,err=*next
    _obj'configPerformed(from "ecStyleCheck",inActionEventID)
return

! The update routine will be executed by the ProvideX event manager during the
! post process logic as defined by the plug-in class - since there no override
! of the getEventNotification() method.
do_update:
enter aPvxState
    _obj'update(from "ecStyleCheck",aPvxState)
return 0

end
```