Inet_Security Function

Description

The function that is called by **OECGI.EXE** before it calls any Internet procedure specified in the HTTP-request.

Note: OpenInsight ships with the source code of this function, as a stored procedure in SYSPROG.

Syntax

return = **Inet_Security** (*Request, InetProcName*)

Parameters

The Inet_Security function has the following parameters:

Parameters	Description
Request	[in] HTTP-request
InetProcName	[in] Name of Internet procedure that is about to be called

Returns

An empty string to allow the call or valid HTML script with an error message; this message is immediately passed back to the client.

See also

Inet_Msg(), Inet_QueryParam(), Inet_Repos(), Inet_Rlist(), Inet_Trace()

Example

To restrict running Inet_ procedures to the SALES user, do the following:

- Log into the **SYSPROG** account.
- The **Inet_Security()** function source, found in **SYSPROCS**, by default, cannot be edited. To turn on editing, edit the **SYSPROCNAMES** row in the **SYSENV** table. This row contains the names of system procedures. Find the entry for **INET_SECURITY**, delete it, and save the row.
- Log out of OpenInsight.
- Log back into the **SYSPROG** account. The **INET_SECURITY** source can now be edited. Modify the source as shown below, with modifications in bold:

Example

```
function INET_Security(Request, FnName)
********************************************************************************
* Project    : OpenInsight for Internet
*
* Name       : INET_Security
* Description: Security check point for a request
*
* Parameters:
*              Request    [in/out] -- HTTP request (see INET_EQUATES)
*              FnName     [in/out] -- name of a function that is
*                                     about to be executed (starts with
*                                     INET_)
*              returns    [out]    -- emty string to allow request or
*                                     message in html format to stop the
*                                     request
*                                     and pass it back to a client
*
* Note:
*              Parameters "Request" and "Function" are passed by
*              reference which makes it possible to change the function
*              name and/or request parameters on a fly by intelligent
*              security procedure
********************************************************************************
$insert Logical
$insert Inet_Equates
declare subroutine Send_Event
HtmlMessage = '' ;* assume success
* put your authorization check code here
*
if @username <> 'SALES' then
  HTMLMessage = "<B>You are not authorized to run " : FnName
end else
  HtmlMessage = ''
end

* if authorization failed -- log the failure
if len(HtmlMessage) then
Send_Event(GS_MONITOR$, 'OMNIEVENT', 'OUTCOMING', 'Authorization failed')
end
return HtmlMessage
```

After you compile the changed program, only the SALES user can run Internet procedures.