



pears.ValidateExternalBlobStoreFilesExist



Generated schema reference. Regenerate this page from the SQL unload; keep hand-maintained business notes in the narrative namespace.

Original SQL

```
CREATE PROCEDURE "pears"."ValidateExternalBlobStoreFilesExist"(  
    /* Application Maintained Function / Procedure - DO NOT EDIT*/  
    IN @AlternativeLocation long VARCHAR DEFAULT NULL )  
RESULT( "Class" CHAR(1),"ID" CHAR(20),"ExternalFilePath" long  
VARCHAR,"Issue" SMALLINT,"Description" CHAR(250) )  
BEGIN  
    DECLARE @Folder long VARCHAR;  
    DECLARE @StandardLocation long VARCHAR;  
    -- create temp table to hold errors  
    DECLARE LOCAL TEMPORARY TABLE "BlobStoreCheckIssue"(  
        "Class" CHAR(1) NULL, -- regularise Standard and Alternative locations  
        "ID" CHAR(20) NULL,  
        "ExternalFilePath" long VARCHAR NULL,  
        "Issue" CHAR(10) NULL,) NOT transactional;SET @StandardLocation =  
(SELECT "BlobExternalRootFolder" FROM "params");  
    IF @AlternativeLocation IS NULL THEN SET @AlternativeLocation =  
@StandardLocation END IF;  
    IF "right"(@StandardLocation,1) <> '\\' THEN SET @StandardLocation =  
"string"(@StandardLocation,'\\') END IF;  
    IF "right"(@AlternativeLocation,1) <> '\\' THEN SET @AlternativeLocation =  
"string"(@AlternativeLocation,'\\') END IF;  
    -- loop through external blobs  
    FOR "BlobLoop" AS "BlobCursor" no scroll cursor FOR  
        SELECT "Class" AS "BClass","ID" AS  
"BID","replace"("ExternalFilePath",@StandardLocation,@AlternativeLocation)  
AS "BExtPath" FROM "BlobStore" WHERE "ExternalFilePath" IS NOT NULL ORDER BY  
"Class" ASC,"ID" ASC FOR READ ONLY  
    do  
        IF(SELECT "byte_substr"("xp_read_file"("BExtPath",1),0,1)) IS NULL THEN  
// ie error reading file  
            INSERT INTO "BlobStoreCheckIssue"(  
"Class","ID","ExternalFilePath","Issue" ) VALUES(  
"BClass","BID","BExtPath",1 )  
        END IF END FOR;  
    -- recheck errors in case files were in use
```



```
FOR "BlobLoop2" AS "BlobCursor2" no scroll cursor FOR
    SELECT "Class" AS "BClass","ID" AS "BID","ExternalFilePath" AS
    "BExtPath" FROM "BlobStoreCheckIssue" WHERE "Issue" = 1 ORDER BY "Class"
    ASC,"ID" ASC FOR READ ONLY
do
    IF(SELECT "byte_substr"("xp_read_file"("BExtPath",1),0,1)) IS NULL THEN
// ie error reading file
        UPDATE "BlobStoreCheckIssue" SET "Issue" = 2 WHERE "Class" = "BClass"
AND "ID" = "BID" // still a problem
    ELSE UPDATE "BlobStoreCheckIssue" SET "Issue" = 0 WHERE "Class" =
    "BClass" AND "ID" = "BID" // now OK
    END IF END FOR;
-- check existence of remaining issues
FOR "BlobLoop3" AS "BlobCursor3" no scroll cursor FOR
    SELECT "Class" AS "BClass","ID" AS "BID","ExternalFilePath" AS
    "BExtPath" FROM "BlobStoreCheckIssue" WHERE "Issue" = 2 ORDER BY "Class"
    ASC,"ID" ASC FOR READ ONLY
do
    SET @Folder = NULL;
    SET @Folder = "left"("BExtPath","locate"("BExtPath",'\\',-1)-1);
    IF EXISTS(SELECT * FROM "sp_list_directory"(@Folder,1) WHERE "file_path"
    = "BExtPath" AND "file_type" = 'F') THEN
        UPDATE "BlobStoreCheckIssue" SET "Issue" = 3 WHERE "Class" = "BClass"
AND "ID" = "BID" // file present IN folder - must be locked OR insufficient
rights TO READ
    ELSE UPDATE "BlobStoreCheckIssue" SET "Issue" = 4 WHERE "Class" =
    "BClass" AND "ID" = "BID" // file NOT present
    END IF END FOR;
    SELECT "Class","ID","ExternalFilePath","Issue",CASE WHEN "Issue" = 3 THEN
    'File present but not accessible' WHEN "Issue" = 4 THEN 'File missing' END
    AS "Description" FROM "BlobStoreCheckIssue"
END
GO

COMMENT TO PRESERVE FORMAT ON PROCEDURE
"pears"."ValidateExternalBlobStoreFilesExist" IS
{CREATE PROCEDURE ValidateExternalBlobStoreFilesExist

/* Application Maintained Function / Procedure - DO NOT EDIT*/

( IN @AlternativeLocation long VARCHAR DEFAULT NULL )
RESULT( Class CHAR(1),ID CHAR(20),ExternalFilePath long VARCHAR,Issue
SMALLINT,Description CHAR(250) )
BEGIN
    DECLARE @Folder long VARCHAR;
    DECLARE @StandardLocation long VARCHAR;
    -- create temp table to hold errors
    DECLARE LOCAL TEMPORARY TABLE BlobStoreCheckIssue(
```



```
Class CHAR(1) NULL, -- regularise Standard and Alternative locations
ID CHAR(20) NULL,
ExternalFilePath long VARCHAR NULL,
Issue CHAR(10) NULL,) NOT transactional;SET @StandardLocation = (SELECT
BlobExternalRootFolder FROM params);
IF @AlternativeLocation IS NULL THEN SET @AlternativeLocation =
@StandardLocation END IF;
IF "right"(@StandardLocation,1) <> '\\' THEN SET @StandardLocation =
string(@StandardLocation, '\\') END IF;
IF "right"(@AlternativeLocation,1) <> '\\' THEN SET @AlternativeLocation =
string(@AlternativeLocation, '\\') END IF;
-- loop through external blobs
FOR BlobLoop AS BlobCursor no scroll cursor FOR
SELECT Class AS BClass, ID AS
BID, REPLACE(ExternalFilePath, @StandardLocation, @AlternativeLocation) AS
BExtPath FROM BlobStore WHERE ExternalFilePath IS NOT NULL ORDER BY Class
ASC, ID ASC FOR READ ONLY
do
IF(SELECT byte_substr(xp_read_file(BExtPath,1),0,1)) IS NULL THEN // ie
error reading file
INSERT INTO BlobStoreCheckIssue( Class, ID, ExternalFilePath, Issue )
VALUES( BClass, BID, BExtPath, 1 ) END IF
END FOR;
-- recheck errors in case files were in use
FOR BlobLoop2 AS BlobCursor2 no scroll cursor FOR
SELECT Class AS BClass, ID AS BID, ExternalFilePath AS BExtPath FROM
BlobStoreCheckIssue WHERE Issue = 1 ORDER BY Class ASC, ID ASC FOR READ ONLY
do
IF(SELECT byte_substr(xp_read_file(BExtPath,1),0,1)) IS NULL THEN // ie
error reading file
UPDATE BlobStoreCheckIssue SET Issue = 2 WHERE Class = BClass AND ID =
BID // still a problem
ELSE UPDATE BlobStoreCheckIssue SET Issue = 0 WHERE Class = BClass AND
ID = BID // now OK
END IF
END FOR;
-- check existence of remaining issues
FOR BlobLoop3 AS BlobCursor3 no scroll cursor FOR
SELECT Class AS BClass, ID AS BID, ExternalFilePath AS BExtPath FROM
BlobStoreCheckIssue WHERE Issue = 2 ORDER BY Class ASC, ID ASC FOR READ ONLY
do
SET @Folder = NULL;
SET @Folder = "left"(BExtPath, locate(BExtPath, '\\', -1) - 1);
IF EXISTS(SELECT * FROM sp_list_directory(@Folder, 1) WHERE file_path =
BExtPath AND file_type = 'F') THEN
UPDATE BlobStoreCheckIssue SET Issue = 3 WHERE Class = BClass AND ID =
BID // file present IN folder - must be locked OR insufficient rights TO
READ
```



```
ELSE UPDATE BlobStoreCheckIssue SET Issue = 4 WHERE Class = BClass AND
ID = BID // file NOT present
END IF
END FOR;
SELECT Class, ID, ExternalFilePath, Issue, CASE WHEN Issue = 3 THEN 'File
present but not accessible' WHEN Issue = 4 THEN 'File missing' END AS
Description FROM BlobStoreCheckIssue
END
}
```

From:

<https://iqxusers.co.uk/iqxhelp/> - iqx

Permanent link:

https://iqxusers.co.uk/iqxhelp/doku.php?id=database:procedures:pears_validateexternalblobstorefilesexist

Last update: **2026/08/07 19:24**

