

Logische Dateinamen



Copyright

© Copyright 2010 SAP AG. Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP AG oder deren Vertriebsfirmen angebotenen Softwareprodukte können Softwarekomponenten auch anderer Softwarehersteller enthalten.

Microsoft, Windows, Excel, Outlook und PowerPoint sind eingetragene Marken der Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli und Informix sind Marken oder eingetragene Marken der IBM Corporation.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und anderen Ländern.

Adobe, das Adobe Logo, Acrobat, PostScript und Reader sind Marken oder eingetragene Marken von Adobe Systems Inc. in den USA und/oder anderen Ländern.

Oracle ist eine eingetragene Marke der Oracle Corporation.

UNIX, X/Open, OSF/1 und Motif sind eingetragene Marken der Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame und MultiWin sind Marken oder eingetragene Marken von Citrix Systems, Inc.

HTML, XML, XHTML und W3C sind Marken oder eingetragene Marken des W3C®, World Wide Web Consortium, Massachusetts Institute of Technology. Java ist eine eingetragene Marke von Sun Microsystems, Inc. JavaScript ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, Clear Enterprise, SAP BusinessObjects Explorer und weitere im Text erwähnte SAP-Produkte und -Dienstleistungen sowie die entsprechenden Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und anderen Ländern.

Business Objects und das Business-Objects-Logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius und andere im Text erwähnte Business-Objects-Produkte und -Dienstleistungen sowie die entsprechenden Logos sind Marken der SAP France in den USA und anderen Ländern.

Alle anderen Namen von Produkten und Dienstleistungen sind Marken der jeweiligen Firmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.

In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden. Die vorliegenden Angaben werden von SAP AG und ihren Konzernunternehmen („SAP-Konzern“) bereitgestellt und dienen ausschließlich Informationszwecken. Der SAP-Konzern übernimmt keinerlei Haftung oder Garantie für Fehler oder Unvollständigkeiten in dieser Publikation. Der SAP-Konzern steht lediglich für Produkte und Dienstleistungen nach der Maßgabe ein, die in der Vereinbarung über die jeweiligen Produkte und Dienstleistungen ausdrücklich geregelt ist. Aus den in dieser Publikation enthaltenen Informationen ergibt sich keine weiterführende Haftung.

Symbole im Text

Symbol	Bedeutung
	Achtung
	Beispiel
	Hinweis
	Empfehlung
	Syntax

In der SAP-Dokumentation werden weitere Symbole verwendet, die verdeutlichen, welche Art von Informationen ein Text enthält. Weitere Informationen finden Sie auf der Startseite jeder Version der SAP-Bibliothek unter *Hilfe zur Hilfe* → *Allgemeine Informationsklassen* und *Informationsklassen für das Business Information Warehouse*.

Typografische Konventionen

Format	Beschreibung
<i>Beispieltext</i>	Wörter oder Zeichen, die vom Bildschirmbild zitiert werden. Dazu gehören Feldbezeichner, Bildtitel, Drucktastenbezeichner sowie Menünamen, Menüpfade und Menüeinträge. Querverweise auf andere Dokumentationen
Beispieltext	hervorgehobene Wörter oder Ausdrücke im Fließtext, Titel von Grafiken und Tabellen
BEISPIELTEXT	Namen von Systemobjekten. Dazu gehören Reportnamen, Programmnamen, Transaktionscodes, Tabellennamen und einzelne Schlüsselbegriffe einer Programmiersprache, die von Fließtext umrahmt sind, wie z.B. SELECT und INCLUDE
Beispieltext	Ausgabe auf der Oberfläche. Dazu gehören Datei- und Verzeichnisnamen und ihre Pfade, Meldungen, Quelltext, Namen von Variablen und Parametern sowie Namen von Installations-, Upgrade- und Datenbankwerkzeugen.
Beispieltext	exakte Benutzereingabe. Dazu gehören Wörter oder Zeichen, die Sie genau so in das System eingeben, wie es in der Dokumentation angegeben ist.
<Beispieltext>	variable Benutzereingabe. Die Wörter und Zeichen in spitzen Klammern müssen Sie durch entsprechende Eingaben ersetzen, bevor Sie sie in das System eingeben.
BEISPIELTEXT	Tasten auf der Tastatur, wie z.B. die Funktionstaste F2 oder die ENTER-Taste

Logische Dateinamen	5
Überblick.....	5
Definition logischer Dateinamen	6
Funktionsbaustein FILE_GET_NAME	9
Funktionsbaustein FILE_VALIDATE_NAME	13
Definition und Verwendung von Aliassen	18
Logische Dateinamen in Programmen verwenden	20

Logische Dateinamen

Viele Anwendungen verwenden Dateien zur Ablage oder Kommunikation und greifen über die ABAP-Befehle `open dataset`, `transfer dataset` usw. darauf zu. Dabei benötigen ABAP-Befehle physische Dateinamen in der Syntax des jeweiligen Betriebssystems. Sie können physische Dateinamen mit logischen Dateinamen verknüpfen. Dadurch können Sie plattformunabhängig Anwendungen schreiben und den Benutzern bestimmte Dateinamen oder Verzeichnisse vorgeben.

Überblick

Folgende typische Szenarios verwenden logische Dateinamen:

- Die Anwendung gibt den logischen Dateinamen im Coding oder Customizing vor. Aus dem logischen Dateinamen ermittelt die Anwendung mit [FILE_GET_NAME \[Seite 9\]](#) einen physischen Dateinamen.
- Die Anwendung lässt die Angabe eines logischen Dateinamens über die Benutzungsoberfläche zu. Die für diesen Kontext zulässigen logischen Dateinamen können über Aliasse konfiguriert werden (siehe [Definition von Aliassen \[Seite 18\]](#)). Der angegebene logische Dateiname wird mit FILE_GET_NAME in einen physischen Dateinamen übersetzt.
- Die Anwendung lässt die Angabe eines physischen Dateinamens über die Benutzungsoberfläche zu. Die Anwendung prüft diesen physischen Dateinamen mit [FILE_VALIDATE_NAME \[Seite 13\]](#) gegen einen in der Anwendung festgelegten logischen Dateinamen. Auch in diesem Zusammenhang werden Aliasse unterstützt.

Die logischen Dateinamen und Pfade definieren Sie im Customizing. Dazu können Sie im Einführungsleitfaden den Abschnitt *SAP NetWeaver → Application Server → Systemadministration → Plattformunabhängige Dateinamen* verwenden oder die Transaktionen **FILE** und **SF01**.

Die von SAP-Anwendungen genutzten Definitionen werden mit dem System ausgeliefert und bei der Einführung ggf. angepasst. Welche Anwendung welche logischen Dateinamen nutzt, entnehmen Sie der anwendungsspezifischen Dokumentation. Sie können weitere Definitionen hinzufügen.

Auslieferung vordefinierter logischer Dateinamen

Für jeden Zugriff, bei dem Benutzer in einem Selektionsbild einen physischen oder logischen Dateinamen angeben können, liefert SAP einen vordefinierten logischen Dateinamen aus.

Das System kann den vom Benutzer angegebenen physischen oder logischen Dateinamen erst prüfen, wenn eine der folgenden Bedingungen erfüllt ist:

- Sie ordnen dem ausgelieferten logischen Dateinamen einen physischen Pfad und/oder Dateinamen zu.
- Sie ordnen dem ausgelieferten logischen Dateinamen ein Alias zu, für welches ein physischer Pfad und/oder Dateiname definiert wurde.

Wenn eine dieser Bedingungen erfüllt ist, ist die Validierung für den entsprechenden vordefinierten logischen Dateinamen aktiv.



In der Regel liefert SAP nur die Definition logischer Dateinamen; die physischen Pfade muss die Systemadministration angeben. Nur wenn physische Pfade

angegeben sind, kann das System Dateinamen prüfen und damit die Sicherheit gewährleisten.

Solange die Validierung für einen ausgelieferten logischen Dateinamen nicht aktiv ist, erzeugt die entsprechende Anwendung bei der Ausführung einen Eintrag im Security Audit Log. Im Security Audit Log können Sie feststellen, welche Zugriffe auf das Dateisystem des Anwendungsservers noch nicht abgesichert sind.



Definition logischer Dateinamen

Mehrstufige, in Tabellen hinterlegte Definitionen steuern die Konvertierung eines logischen Dateinamens in einen plattformspezifischen physischen Dateinamen.

Definition des logischen Dateinamens

- **Logischer Dateiname**

Plattformunabhängiger sprechender Name einer Datei. Das System legt ihn zwar grundsätzlich mandantenübergreifend an, aber Sie können ihn zusätzlich auch mandantenspezifisch definieren. Logische Dateinamen dienen zum einen der plattformunabhängigen Programmierung und zum anderen der Validierung von Dateinamen.

- **Physischer Dateiname**

Plattformabhängiger Name einer Datei. Zu einem logischen Dateinamen ist pro Syntaxgruppe des Betriebssystems genau ein physischer Dateiname hinterlegt.

- **Datenformat**

Beschreibt das Format der Daten.



Neben den üblichen Datenformaten wie ASCII oder dem binären Datenformat können Sie auch das Format **DIR** auswählen. Mit dem Format **DIR** legen Sie fest, dass der zugehörige logische Dateiname keine gewöhnliche Datei bezeichnet, sondern ein **Prüfverzeichnis zur Validierung von Dateinamen**. Weitere Informationen über das Arbeiten mit dem Format **DIR** finden Sie unter [Funktionsbaustein FILE_VALIDATE_NAME \[Seite 13\]](#) im Abschnitt zum Importparameter `INCLUDING_DIR`.

- **Arbeitsgebiet**

Beschreibt die inhaltliche Zugehörigkeit der betreffenden Datei

- **Logischer Pfad**

Ein logischer Pfad ist ein plattformunabhängiger sprechender Name für einen Pfad. Zu einem logischen Dateinamen ist in der Regel auch ein logischer Pfad hinterlegt; nur dann können für den logischen Dateinamen in Abhängigkeit von der jeweiligen Betriebssystemplattform unterschiedliche physische Pfade erzeugt werden.

Definition des logischen Pfades

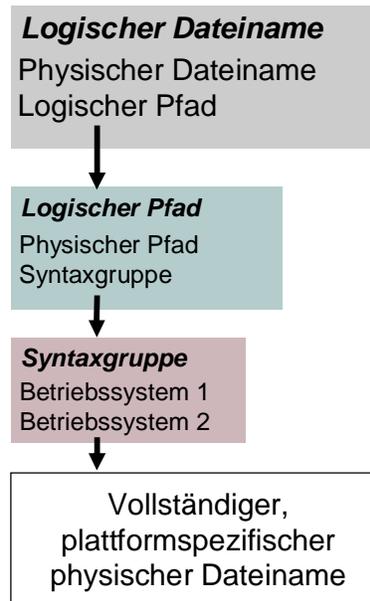
- **Physischer Pfad**

Plattformabhängiger Name eines logischen Pfades für eine bestimmte Syntaxgruppe. Zu einem logischen Pfad sind für eine oder mehrere Syntaxgruppen die entsprechenden plattformspezifischen physischen Pfade hinterlegt.

- **Betriebssysteme und Syntaxgruppen**

Alle in einer Systemkonfiguration verwendeten Betriebssysteme sind Syntaxgruppen zugeordnet. Eine Syntaxgruppe ist eine Sammelbezeichnung für eine Gruppe von Betriebssystemen, die eine gemeinsame Syntax für Dateinamen und Pfade haben. Ihre Definition legt u.a. die maximale Länge der Dateinamen und die Zulässigkeit von Dateinamenerweiterungen fest.

Folgende Abbildung zeigt die Beziehungen zwischen den Objekten, welche die Umsetzung eines logischen in einen physischen Dateinamen bestimmen:



Parameter in physischen Dateinamen und Pfaden

Physische Dateinamen und Pfade können, in spitze Klammern eingefasst, folgende reservierte Wörter als Platzhalter enthalten. Das System ersetzt die Platzhalter zur Laufzeit durch aktuelle Werte.

Reservierte Wörter

Reserviertes Wort	Ersetzungswert
<OPSYS>	Betriebssystem laut Funktionsbaustein-Parameter (s. unten)
<INSTANCE>	Instanz der Anwendung
<SYSID>	Name der Anwendung laut Systemfeld SY-SYSID.
<DBSYS>	Datenbanksystem laut Systemfeld SY-DBSYS
<SAPRL>	Release laut Systemfeld SY-SAPRL
<HOST>	Rechnername laut Systemfeld SY-HOST
<CLIENT>	Mandant laut Systemfeld SY-MANDT
<LANGUAGE>	Anmeldesprache laut Systemfeld SY-LANGU
<DATE>	Datum laut Systemfeld SY-DATUM
<YEAR>	Jahr laut Systemfeld SY-DATUM, vierstellig
<YEAR>	Jahr laut Systemfeld SY-DATUM, zweistellig
<MONTH>	Monat laut Systemfeld SY-DATUM
<DAY>	Tag laut Systemfeld SY-DATUM
<WEEKDAY>	Wochentag laut Systemfeld SY-FDAYW
<TIME>	Uhrzeit laut Systemfeld SY-UZEIT
<STIME>	Stunde und Minute laut Systemfeld SY-UZEIT
<HOUR>	Stunde laut Systemfeld SY-UZEIT
<MINUTE>	Minute laut Systemfeld SY-UZEIT
<SECOND>	Sekunde laut Systemfeld SY-UZEIT
<PARAM_1>	beim Funktionsaufruf übergebener externer Parameter 1
<PARAM_2>	beim Funktionsaufruf übergebener externer Parameter 2
<PARAM_3>	beim Funktionsaufruf übergebener externer Parameter 3
<P=name>	Wert eines Profilparameters des laufenden Systems
<V=name>	Wert einer Variablen gemäß Variablen-tabelle
<F=name>	Rückgabewert eines Funktionsbausteins



Alle physischen Pfade müssen das reservierte Wort <FILENAME> als Platzhalter für den physischen Dateinamen enthalten.

Wenn Sie solche Parameter in physische Datei- bzw. Pfadnamen aufnehmen, unterstützen Sie eine differenzierte und zugleich standardisierte Vergabe von Dateinamen. Wenn Sie eine logische Datei in kurzer Zeit mehrfach sichern, ist z.B. der Parameter <TIME> sinnvoll. Neben den Systemfeldwerten können Sie die Namensvergabe flexibel gestalten, insbesondere wenn Sie die zuletzt aufgeführten Parameter verwenden:

- Mit <PARAM_1> bis <PARAM_3> können Werte in die Dateinamen bzw. Pfade aufgenommen werden, die im Anwendungsprogramm beim Aufruf des Funktionsbausteins FILE_GET_NAME explizit übergeben werden.
- Mit <P=name> können Sie die Werte von Profilparametern des laufenden Systems aufnehmen. Wenn Sie den Report RSPARAM ausführen, erhalten Sie eine Liste der Profilparameter und ihrer Werte.

- Mit <V=name> können Sie die Werte von Variablen aufnehmen, die Sie mit Transaktion FILE in den Steuerungstabellen definiert haben.
- Mit <F=name> können Sie Rückgabewerte von Funktionsbausteinen aufnehmen. Der Name eines hier verwendeten Funktionsbausteins muss mit dem Präfix "FILENAME_EXIT_" beginnen. Beachten Sie, dass der Funktionsbaustein im reservierten Wort lediglich mit dem Teil seines Namens adressiert wird, der auf dieses Präfix folgt. Wenn Sie z.B. Funktionsbaustein FILENAME_EXIT_EXAMPLE verwenden, lautet der Platzhalter <F=EXAMPLE>. Der Funktionsbaustein muss den Exportparameter OUTPUT haben, und Sie dürfen für diesen keinen Bezugstyp angeben. Importparameter müssen mit Vorschlagswerten belegt sein. Tabellenparameter werden nicht unterstützt.



Funktionsbaustein FILE_GET_NAME

Anwendungsprogramme nutzen logische Dateinamen in Verbindung mit dem Funktionsbaustein FILE_GET_NAME. Basierend auf den im Customizing hinterlegten Definitionen erzeugt der Funktionsbaustein zur Laufzeit zu einem logischen Dateinamen den entsprechenden physischen Dateinamen.

Die folgende Tabelle gibt eine Übersicht der Import- und Exportparameter sowie der Ausnahmereingungen des Funktionsbausteins.

Schnittstellen des Funktionsbausteins FILE_GET_NAME

Importparameter	Funktion
CLIENT	Logische Dateinamen können auch mandantenspezifisch definiert sein. Dieser Parameter ermittelt den Mandanten, dessen Definition der Funktionsbaustein verwenden soll. Der Vorschlagswert ist der aktuelle Mandant, wie er im Systemfeld SY-MANDT abgelegt ist.
LOGICAL_FILENAME	Übergibt den logische Dateinamen. (Großschreibung ist erforderlich!)
OPERATING_SYSTEM	Bestimmt das Betriebssystem, für das der Dateiname ermittelt werden soll. Vorschlagswert ist das Betriebssystem des Anwendungsservers, wie es im Systemfeld SY OPSYS abgelegt ist.
PARAMETER_1 PARAMETER_2 PARAMETER_3	Kann beliebige Werte an die mit <PARAM_1> bis <PARAM_3> bezeichneten Platzhalter in den physischen Datei- und Pfadnamen übergeben.
USE_PRESENTATION_SERVER	Gibt an, dass der Dateiname auf der Grundlage des Betriebssystems des Präsentationsservers ermittelt werden soll. Das durch Parameter OPERATING_SYSTEM importierte Betriebssystem soll nicht verwendet werden.
WITH_FILE_EXTENSION	Gibt an, dass das für den logischen Dateinamen definierte Dateiformat als Dateinamenerweiterung an den physischen Dateinamen angehängt werden soll. Ausnahme: Wenn das Datenformat des logischen Dateinamens DIR ist, ignoriert das System den Parameter WITH_FILE_EXTENSION (siehe Parameter INCLUDING_DIR).
USE_BUFFER	Gibt an, dass die Steuerungstabellen gepuffert werden sollen.

ELEMIMATE_BLANKS	Legt fest, ob sämtliche Leerzeichen aus dem zurückgegeben Dateinamen entfernt werden sollen.
INCLUDING_DIR	<p>In den meisten Anwendungsfällen benötigt die Anwendung, welche den Funktionsbaustein FILE_GET_NAME aufruft, einen vollständigen physischen Dateinamen zur weiteren Verarbeitung, z.B. zur Verwendung in Verbindung mit OPEN DATASET oder ähnlichen Befehlen.</p> <p>Für die Validierung von Benutzereingaben wurde das Datenformat DIR eingeführt. Wenn ein logischer Dateiname das Datenformat DIR hat, ist das Ergebnis von FILE_GET_NAME ein physischer Pfadname.</p> <p>Standardmäßig ist das Kennzeichen INCLUDING_DIR nicht gesetzt. Das aufrufende Programm gibt somit an, dass es einen physischen Dateinamen benötigt. Da ein logischer Dateiname mit dem Datenformat DIR diese Bedingung nicht erfüllt, löst das System die Ausnahme FILE_NOT_FOUND aus.</p> <p>Wenn die aufrufende Anwendung das Kennzeichen INCLUDING_DIR setzt, gibt sie damit an, dass sie als Ergebnis von FILE_GET_NAME auch einen physischen Dateipfad verarbeiten kann. In diesem Fall wird der ermittelte Pfadname an die aufrufende Anwendung zurückgegeben.</p> <p></p> <p>Wenn das Kennzeichen INCLUDING_DIR gesetzt ist, ignoriert das System den Parameter WITH_FILE_EXTENSION. Es gibt eventuell einen leeren Dateinamen an die aufrufende Anwendung zurück. Es führt in diesem Fall die Logik gemäß der Beschreibung des Exportparameters EMERGENCY_FLAG nicht aus.</p> <p>Weitere Informationen über die Validierung von Benutzerangaben finden Sie unter Funktionsbaustein FILE_VALIDATE_NAME [Seite 13] im Abschnitt Prüfung auf Verzeichnisse.</p>

Exportparameter	Funktion
EMERGENCY_FLAG	Wenn der Wert ungleich SPACE ist, konnte das System zum logischen Dateinamen für das vorliegende Betriebssystem aus den Angaben im Customizing keinen physischen Dateinamen ermitteln (weitere Details finden Sie im Hinweis im Anschluss an die Schnittstellendokumentation). In diesem Fall verwendet das System den im Profilparameter DIR_GLOBAL angegebenen Pfad als physischen Pfad.
FILE_FORMAT	Dieser Parameter enthält das für den logischen Dateinamen definierte Dateiformat. Sie können mit diesem Parameter z.B. entscheiden, in welchem Modus die Datei geöffnet werden soll.
FILE_NAME	Dieser Parameter enthält den vollständigen physischen Dateinamen einschließlich Pfadangabe.

Ausnahmen	Funktion
FILE_NOT_FOUND	Wird ausgelöst, wenn der logische Dateiname nicht definiert ist.
OTHERS	Wird ausgelöst, wenn andere Fehler auftreten.



Die folgenden Ursachen führen dazu, dass der Funktionsbaustein für das vorliegende Betriebssystem keinen physischen Pfad ermitteln kann (siehe Parameter EMERGENCY_FLAG):

- Das Betriebssystem ist in den Steuerungstabellen nicht definiert.
- Das Betriebssystem ist zwar definiert, aber keiner Syntaxgruppe zugeordnet.
- Dem logischen Pfad ist für die Syntaxgruppe kein physischer Pfad zugeordnet.
- Dem logischen Dateinamen ist kein logischer Pfad zugeordnet.

Beispiel

Zur logischen Datei DATA_FILE und dem logischen Pfad DATA_PATH liegen folgende Definitionen vor:

DATA_FILE	phys. Datei:	datei<PARAM_1>	
	Datenformat:	BIN	
	logischer Pfad:	DATA_PATH	
DATA_PATH	Syntaxgruppe:	UNIX	phys. Pfad: /tmp/<FILENAME>
	Syntaxgruppe:	Windows NT	phys. Pfad: c:\tmp\<FILENAME>

Das Betriebssystem des Anwendungsservers ist der Syntaxgruppe UNIX, das Betriebssystem des Präsentationsservers der Syntaxgruppe Windows NT zugeordnet. Die beiden folgenden Aufrufe des Funktionsbausteins liefern die jeweils angegebenen Rückgabewerte.

Beispiel 1

```

CALL FUNCTION 'FILE_GET_NAME'

  EXPORTING
    LOGICAL_FILENAME      = 'DATA_FILE'
    PARAMETER_1           = '01'

  IMPORTING
    EMERGENCY_FLAG        = LD_EMERGENCY_FLAG
    FILE_FORMAT            = LD_FILE_FORMAT
    FILE_NAME              = LD_FILE_NAME

  EXCEPTIONS
    FILE_NOT_FOUND        = 1
    OTHERS                 = 2.

```

Rückgabewerte:

Variable	Inhalt
ld_emergency_flag	Space
ld_file_format	BIN
ld_file_name	/tmp/datei01

Beispiel 2

```

CALL FUNCTION 'FILE_GET_NAME'

  EXPORTING
    LOGICAL_FILENAME      = 'DATA_FILE'
    USE_PRESENTATION_SERVER = 'X'
    WITH_FILE_EXTENSION   = 'X'

  IMPORTING
    EMERGENCY_FLAG        = LD_EMERGENCY_FLAG
    FILE_FORMAT            = LD_FILE_FORMAT
    FILE_NAME              = LD_FILE_NAME

  EXCEPTIONS
    FILE_NOT_FOUND        = 1
    OTHERS                 = 2.

```

Rückgabewerte:

Variable	Inhalt
ld_emergency_flag	Space
ld_file_format	BIN
ld_file_name	C:\tmp\DATEI.BIN

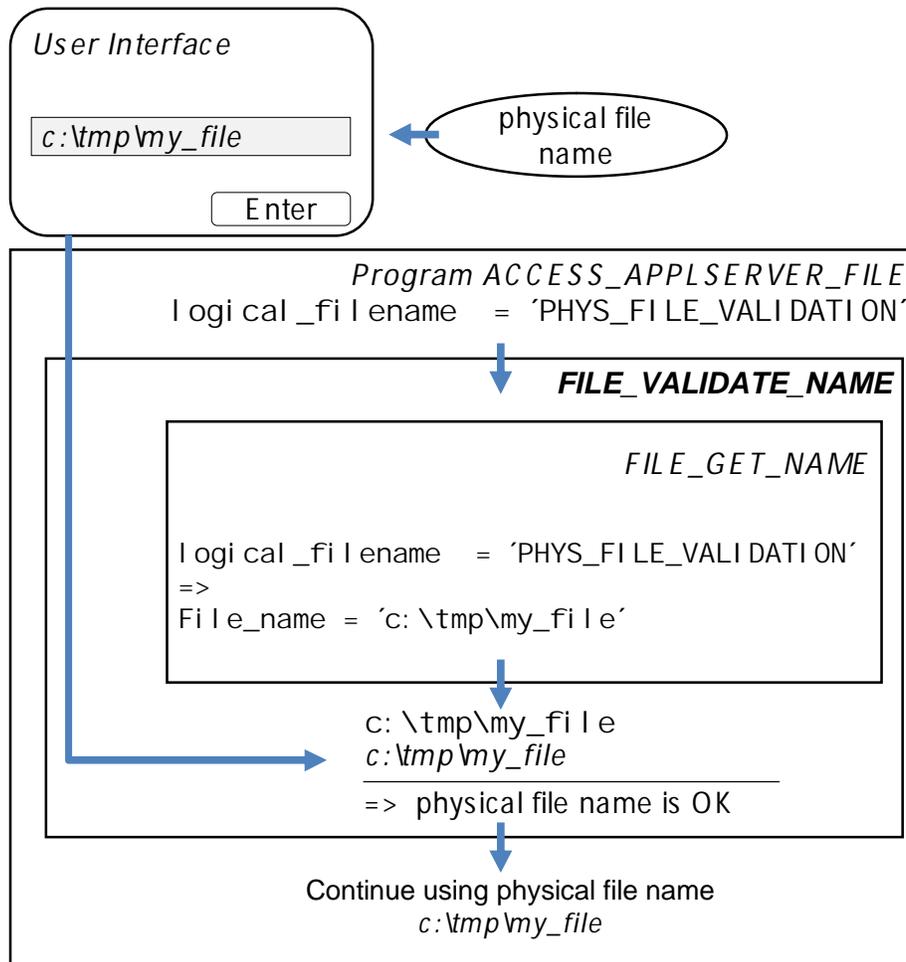


Funktionsbaustein FILE_VALIDATE_NAME

Der Funktionsbaustein FILE_VALIDATE_NAME prüft mit einem logischen Dateinamen, ob ein physischer Dateiname den Vorgaben des logischen Dateinamens entspricht.

Beispiel

Folgende Abbildung zeigt an einem vereinfachten Beispiel das Funktionsprinzip von FILE_VALIDATE_NAME. In diesem Beispiel ist das Betriebssystem des Anwendungsservers Microsoft Windows.



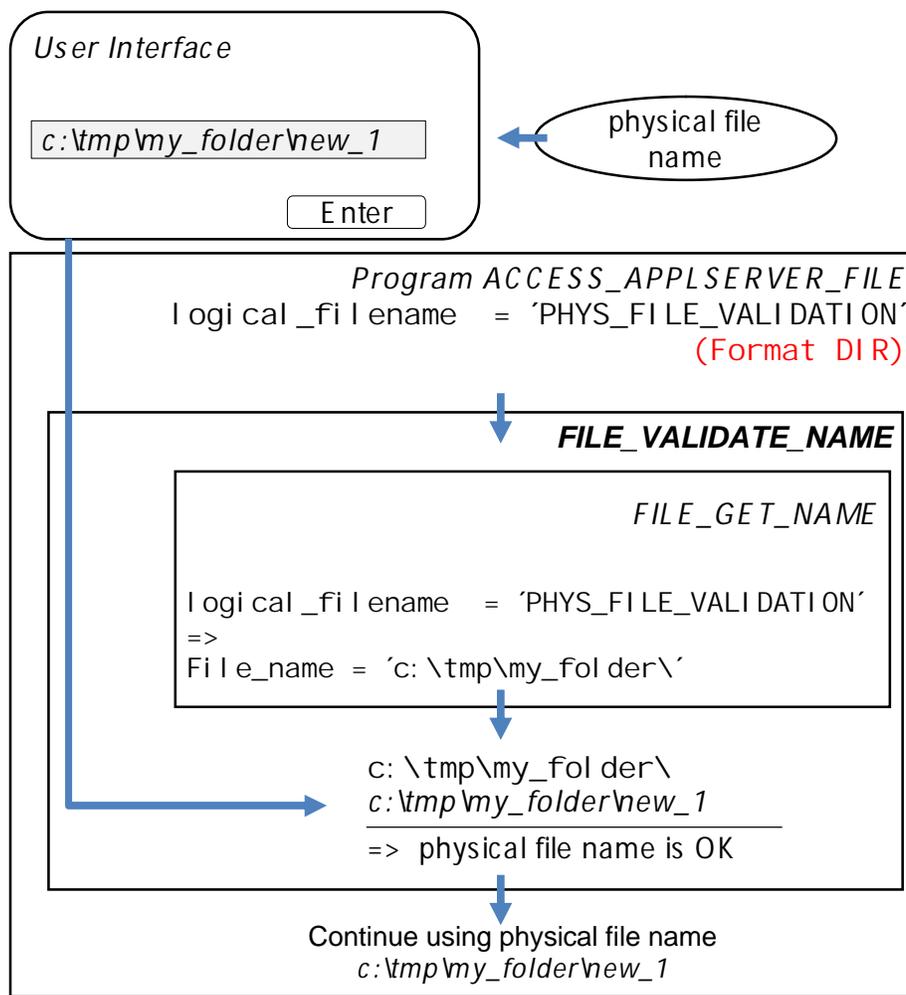
Das Beispielprogramm ACCESS_APPLSERVER_FILE kennt als konstanten Wert den logischen Dateinamen *PHYS_FILE_VALIDATION*.

Wenn ein Benutzer mit dem Programm auf die Datei *c:\tmp\my_file* zugreift, laufen folgende Schritte ab:

- Das System ruft Funktionsbaustein FILE_VALIDATE_NAME auf.
- Dieser ruft Funktionsbaustein FILE_GET_NAME auf. Damit ermittelt er den zum logischen Dateinamen und zum passenden Betriebssystem zugehörigen vollständigen physischen Dateinamen. Diesen gibt er an FILE_VALIDATE_NAME zurück.
- Das System vergleicht die beiden physischen Dateinamen – aus der Selektionsbildeingabe einerseits und der Rückgabe des FILE_GET_NAME-Aufrufs andererseits – miteinander.
- Wenn die beiden Dateinamen übereinstimmen, kann der Benutzer auf die Datei zugreifen.
- Wenn die beiden Dateinamen nicht übereinstimmen, löst Funktionsbaustein FILE_VALIDATE_NAME die Ausnahme VALIDATION_FAILED aus. Das Programm kann dem Benutzer den Zugriff auf die Datei verweigern.

Prüfung auf Verzeichnisse

Sie brauchen die Prüfung nicht auf konkrete Dateinamen zu beschränken, sondern können ganze Bereiche eines Dateisystems als Zugriffsziel zulassen. In diesem Fall führt das System die Prüfung nur für den Pfad des hinterlegten Verzeichnisses aus:

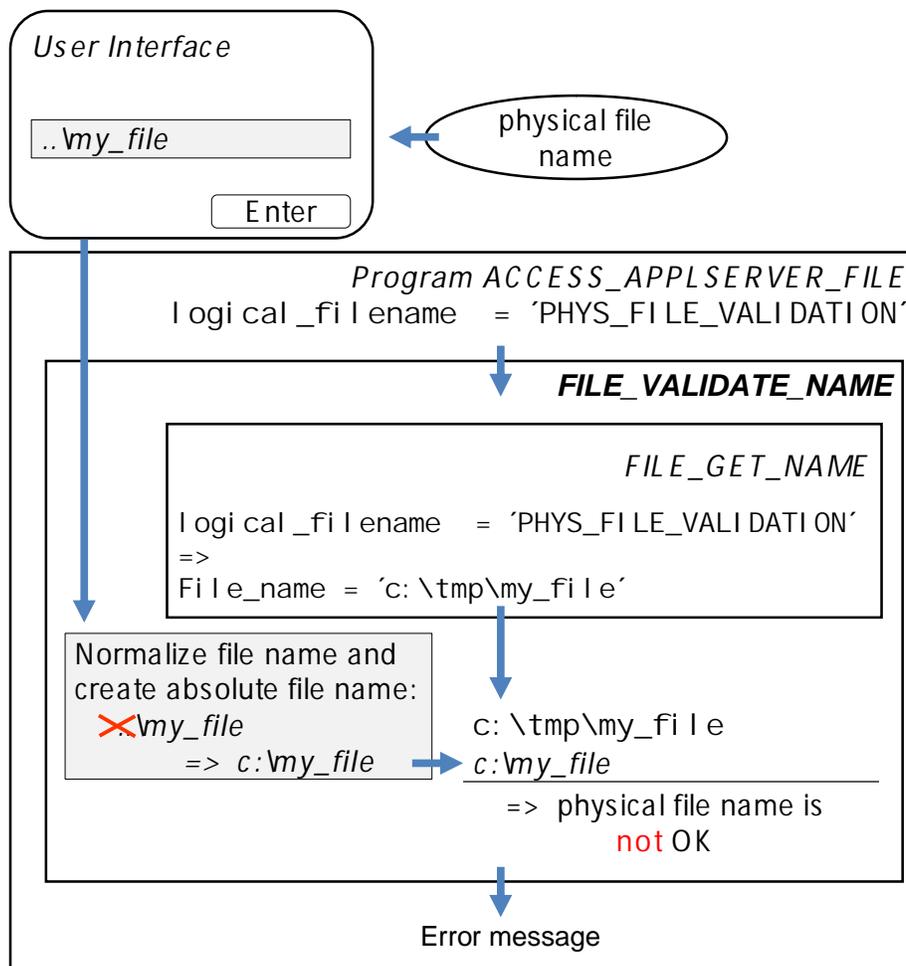


Voraussetzung hierfür ist, dass der logische Dateiname das Datenformat **DIR** hat.

Normalisierung und absolute Dateinamen

Funktionsbaustein `FILE_VALIDATE_NAME` übernimmt noch eine weitere Funktion.

Wenn ein Benutzer einen physischen Dateinamen eingegeben hat, welcher Betriebssystemkommandos zur Navigation im Dateisystem enthält, dann interpretiert das System diese Navigation vor dem Vergleich mit dem Ergebnis aus `FILE_GET_NAME`. Ebenso wandelt es den ermittelten physischen Dateinamen in einen absoluten Dateinamen um, um einen sinnvollen Vergleich zu ermöglichen. Im folgenden Beispiel ist das globale Arbeitsverzeichnis `c:\tmp`.



Auf diese Weise ist sichergestellt, dass der Benutzer nur auf in diesem Kontext vorgesehene Dateien zugreifen kann.

Schnittstellen des Funktionsbausteins FILE_VALIDATE_NAME

Importparameter	Funktion
CLIENT	Logische Dateinamen können auch mandantenspezifisch definiert sein. Dieser Parameter ermittelt den Mandanten, dessen Definition das System verwenden soll. Vorschlagswert ist der aktuelle Mandant, wie er im Systemfeld SY-MANDT abgelegt ist.
LOGICAL_FILENAME	Logischer Dateiname, der für die Validierung verwendet werden soll (Großschreibung ist erforderlich).
OPERATING_SYSTEM	Bestimmt das Betriebssystem, für das der Dateiname ermittelt werden soll. Vorschlagswert ist das Betriebssystem des Anwendungsservers, wie es im Systemfeld SY OPSYS abgelegt ist.
PARAMETER_1 PARAMETER_2 PARAMETER_3	Können beliebige Werte an die mit <PARAM_1> bis <PARAM_3> bezeichneten Platzhalter in den physischen Datei- und Pfadnamen übergeben.
WITH_FILE_EXTENSION	Kennzeichnet, dass das für den logischen Dateinamen definierte Dateiformat als Dateinamenerweiterung an den physischen Dateinamen angehängt werden soll.
USE_BUFFER	Kennzeichnet, dass die Steuerungstabellen gepuffert werden sollen.
ELIMINATE_BLANKS	Legt fest, ob alle Leerzeichen aus dem zurückgegeben Dateinamen entfernt werden sollen.



Die Importparameter werden von FILE_VALIDATE_NAME an FILE_GET_NAME weitergegeben, um einen physischen Dateinamen zu ermitteln. Mit diesem vergleicht das System den Inhalt des CHANGING-Parameters PHYSICAL_FILENAME. Sie beeinflussen also die Bestimmung des physischen Dateinamens, der für die Validierung verwendet wird.

Exportparameter	Funktion
VALIDATION_ACTIVE	Wenn der Wert ungleich SPACE ist, konnte das System zum logischen Dateinamen oder einem zugeordneten Alias einen physischen Dateinamen ermitteln.
TS_ALIAS	Sortierte Tabelle der definierten Aliasse (einschließlich des übergebenen logischen Dateinamens).

CHANGING-Parameter	Funktion
PHYSICAL_FILENAME	Beschreibt den physischen Dateinamen, den FILE_VALIDATE_NAME mit den Einstellungen des übergebenen logischen Dateinamens vergleicht. Da der physische Dateiname ggf. normalisiert werden muss, ist er vom Typ CHANGING.

AUSNAHMEN	Funktion
LOGICAL_FILENAME_NOT_FOUND	Wird ausgelöst, wenn der Funktionsbaustein FILE_GET_NAME seinerseits für alle Aliasse die Ausnahme FILE_NOT_FOUND an FILE_VALIDATE_NAME propagiert hat.
VALIDATION_FAILED	Wird ausgelöst, wenn der physische Dateiname in PHYSICAL_FILENAME nicht den Vorgaben gemäß LOGICAL_FILENAME entspricht.



Definition und Verwendung von Aliassen

Um die Wiederverwendung logischer Dateinamen bei der Validierung von Dateinamen zu ermöglichen und die Validierung flexibler zu gestalten, können Sie Aliasse für logische Dateinamen definieren. Ein Alias ist selbst ein logischer Dateiname.



Beispiel für die Prüfung eines physischen Dateinamens

Im Programm ACCESS_APPLSERVER_FILE soll der Benutzer den physischen Dateinamen `/usr/sap/PRD_100/tmp/my_file` angeben. Das Programm ACCESS_APPLSERVER_FILE verwendet für die Validierung den logischen Dateinamen PHYS_FILE_VALIDATION, welchem im Auslieferungszustand kein physischer Pfad zugeordnet ist. Sie haben in Ihrem System bereits einen anderen logischen Dateinamen Z_FILE_TMP, welcher den physischen Dateinamen `/usr/sap/PRD_100/tmp/my_file` erzeugt. Statt diese Zuordnung für den logischen Dateinamen PHYS_FILE_VALIDATION ein weiteres Mal zu erzeugen, tragen Sie Z_FILE_TMP als Alias für PHYS_FILE_VALIDATION ein.



Beispiel für die Prüfung physischer Dateipfade

Im Programm ACCESS_APPLSERVER_FILE2 soll der Benutzer einen physischen Dateinamen im Verzeichnissbereich `/usr/sap/PRD_100/tmp/` oder `/usr/sap/PRD_100/work/` angeben. Das Programm ACCESS_APPLSERVER_FILE2 verwendet für die Validierung den logischen Dateinamen PHYS_FILE_VALIDATION2, welchem im Auslieferungszustand kein physischer Pfad zugeordnet ist. Sie definieren zwei neue logische Dateinamen Z_DIR_TMP und Z_DIR_WORK, welche das Datenformat DIR haben und denen die zuvor genannten Verzeichnispfade zugeordnet sind. Sie tragen Z_DIR_TMP und Z_DIR_WORK als Aliasse für PHYS_FILE_VALIDATION2 ein.



Beispiel für die Angabe eines logischen Dateinamens an der Benutzungsoberfläche

Im Programm ACCESS_APPLSERVER_LOGFILE soll der Benutzer nur bestimmte logische Dateinamen angeben dürfen. Das Programm ACCESS_APPLSERVER_LOGFILE verwendet für die Validierung den logischen Dateinamen LOG_FILE_VALIDATION, welchem im Auslieferungszustand kein physischer Dateiname zugeordnet ist.

Sie haben in Ihrem System bereits die logischen Dateinamen ZACCESS_APPLSERVER_LOGFILE1 und ZACCESS_APPLSERVER_LOGFILE2 definiert, welche zulässige physische Dateinamen erzeugen. Sie tragen nun die logischen Dateinamen ZACCESS_APPLSERVER_LOGFILE1 und ZACCESS_APPLSERVER_LOGFILE2 als Aliasse für LOG_FILE_VALIDATION ein.

Der Benutzer kann nun den logischen Dateinamen ZACCESS_APPLSERVER_LOGFILE1 und ZACCESS_APPLSERVER_LOGFILE2 für das Programm

ACCESS_APPLSERVER_LOGFILE angeben. Der logische Dateiname LOG_FILE_VALIDATION darf in diesem Beispiel nicht angegeben werden, da diesem logischen Dateinamen kein physischer Dateiname zugeordnet wurde. Wenn der Benutzer diesen logischen Dateinamen ebenfalls angeben können soll, müssen Sie dem logischen Dateinamen LOG_FILE_VALIDATION einen physischen Dateinamen zuordnen.



Es hat bei der Prüfung eines logischen Dateinamens keinen Sinn, einen logischen Dateinamen als Alias zuzuordnen, welcher das Datenformat DIR hat. Das Programm benötigt einen logischen Dateinamen, welcher einen vollständigen physischen Dateinamen definiert.

Verallgemeinerung

Wenn Sie ein Alias für einen logischen Dateinamen definieren, wird das Alias bei der Validierung eines Dateinamens gleichwertig zum ausgelieferten logischen Dateinamen einer Anwendung für die Validierung behandelt. Das heißt ein Dateiname ist gemäß Validierung dann zulässig, wenn er den Vorgaben des ursprünglichen logischen Dateinamens oder eines definierten Alias entspricht.

Pflege von Aliassen

Sie können Aliasse im Einführungsleitfaden (IMG) unter *SAP NetWeaver → Application Server → Systemadministration → Plattformunabhängige Dateinamen → Aliasse für logische Dateinamen pflegen* pflegen.

Dieser IMG-Pfad hängt vom Release und Support-Package-Stand Ihres Systems ab und ist eventuell nicht vorhanden. In diesem Fall können Sie Aliasse mit Transaktion **SM30** über die View **V_FILEALIA** (ab Release 46C und höher: View **V_FILEA31I**) pflegen.

Logische Dateinamen in Programmen verwenden

Um in einem Ihrer Anwendungsprogramme logische Dateinamen zu verwenden, müssen Sie folgende Schritte durchführen:

1. Überprüfen Sie mit Transaktion FILE, ob im System Definitionen zu den logischen Dateinamen hinterlegt sind. Diese Definitionen konvertieren die logischen in plattformspezifische Dateinamen. Legen Sie ggf. mit Transaktion FILE neue Definitionen an.

Weitere Informationen finden Sie im Einführungsleitfaden unter *SAP NetWeaver* → *Application Server* → *Systemadministration* → *Plattformunabhängige Dateinamen*.
2. Vergewissern Sie sich, dass die in diesen Definitionen angegebenen physischen Pfade im Dateisystem der Laufzeitumgebung vorhanden sind. Legen Sie ggf. die erforderlichen Verzeichnisse an, oder wenden Sie sich an die Systemadministration.
3. Testen Sie die Dateinamenkonvertierung durch Aufruf des Funktionsbausteins im Einzeltest der Transaktion SE37.



Setzen Sie beim Einzeltest der Transaktion SE37 das Kennzeichen Klein-Groß-Schreibung, da insbesondere das Feld OPERATING_SYSTEM zwischen Groß- und Kleinbuchstaben unterscheidet. Entsprechend müssen Sie den logischen Dateinamen vollständig in Großbuchstaben angeben.

4. Fügen Sie einen Aufruf des Funktionsbausteins in Ihr Programm ein. Sie können hierzu den Menüpfad *Bearbeiten* → *Muster* verwenden.

Weitere Informationen über das Speichern von Dateien auf dem Anwendungs- und dem Präsentationsserver finden Sie unter *Arbeiten mit Dateien* im *ABAP-Benutzerhandbuch*.



Beachten Sie, dass die Erzeugung eines syntaktisch korrekten logischen Dateinamens alleine noch nicht das reibungslose Abspeichern einer Datei gewährleistet. Der physische Pfad muss zur Laufzeit im Dateisystem auch tatsächlich vorhanden sein.



Lassen Sie auf der Benutzungsoberfläche nicht zu, dass der Benutzer bei der Ausführung einen beliebigen vorhandenen logischen Dateinamen angeben kann. Hinterlegen Sie den logischen Dateinamen fest in der Programmlogik, oder folgen Sie der Anleitung über die Verwendung logischer Dateinamen in der Benutzungsoberfläche. Weitere Informationen finden Sie unter [Secure Programming - ABAP \[Extern\]](#).