



Az informatika nagyvállalati környezetben való megjelenése óta az üzleti területek részéről is természetszerűleg merül fel a szakterületi háttér folyamatok IT eszközökkel való támogatásának igénye. Az adattárházak megjelenésével ezen igények létjogosultsága tovább erősödött azáltal, hogy a szervezetek számára kulcsfontosságú területté vált a vállalati adatvagyon minél hatékonyabb és precízebb tárolása, feldolgozása és kiaknázása. Annak érdekében, hogy a vállalat által birtokolt hatalmas mennyiségű adatból olyan következtetésekre lehessen jutni az üzleti döntéshozatal során, amely hozzájárulhat a szervezeti célok eléréséhez (pl. üzleti döntéshozatali folyamat gyorsítása), elengedhetetlen az adatok pontosságának és megbízhatóságának garantálása.

Cikkünk első részében a nagyvállalati adatkarbantartás kihívásait és általános megvalósítási lehetőségeit járjuk körül, miközben az alábbi kérdésekre igyekszünk választ találni:

- Milyen kihívásokkal szembesülnek a nagyvállalatok az adatkarbantartás során, figyelembe véve az adatok növekvő mennyiségét és sokféleségét, valamint az adatbiztonsággal, integritással összefüggő fenyegetéseket?
- Milyen általános módszerek és technológiák állnak a nagyvállalatok rendelkezésére az adatkarbantartási funkciók hatékonyabbá tételére, beleértve a közvetlen adatmódosítást, az újrafuttatható patchprogramokat és az általános célú alkalmazásokat?
- Hogyan lehet a manualitásból eredő kockázatokat, átfutási időket csökkentve az adatmenedzsmenttel kapcsolatos befektetést stratégiai előnnyé alakítani?

Problémakör bemutatása

Az üzleti döntések alapjául szolgáló információs bázis létrehozásához szükséges bemenő adatok elvart minőségben és határidőre történő előállítás mellett, olyan alkalmazások kialakítására is szükség van, amelyek képesek a komplex adatkarbantartási- és adatmenedzsment folyamatok hatékony támogatására. Ilyen folyamat lehet például egy üzleti döntéstámogató rendszer havi futtatásához szükséges üzleti paraméterek beállítása, amely során meghatározásra kerül többek között a megfigyelési időszak, az ügyfelek, ügyletek meghatározott attribútumaira vonatkozó szűrési kritérium, illetve a feldolgozás során figyelembe veendő tranzakciós adatok részletezettségi szintje. Ezen paraméterezési feladatok végrehajtása az üzleti területek kompetenciájába tartozik, hiszen a nyers

adatok üzleti értelemmel való felruházása ezen a szinten történik, ezen a ponton válik információvá az adat.

Az informatikai rendszerek fejlesztése során, az elmúlt években megfigyelhető olyan törekvés, amelynek fókuszában a 2000-es években bevezetett monolitikus rendszerek modernizációja, funkcionális dekompozíciója áll. Ennek magyarázata többek között a mára elavultnak számító, nem támogatott technológiák kivezetése, az architektúra általános modernizációja, illetve a szállítói kitétség csökkentése. A mikroszolgáltatások bevezetésével robosztus, jól skálázható, hibatűrő és könnyebben karbantartható megoldások hozhatók létre, azonban a lazán csatolt architektúra elemek integrációjának megoldása nemcsak a rendszerüzemeltetés számára jelent kihívást, hanem az alkalmazások működési paramétereit karbantartó üzleti felhasználók számára is. A rendszeres és eseti jellegű paraméterezési feladatok rendszertámogatás nélküli megvalósítása kockázatos és időigényes művelet.

Nagyvállalati környezetben a produktív rendszerek működését befolyásoló, szabályozó paraméterek módosítása csak szigorúan szabályozott, standardizált és auditálható módon valósítható meg, az üzembiztonságra és üzletmenet-folytonosságra vonatkozó elsődleges követelmények miatt, figyelembe véve az adatok konzisztenciájára és védelmére vonatkozó legalapvetőbb elvárásokat. Az emberi tényezőből eredő tévedések lehetőségének minimalizálása érdekében, gyakran alkalmaznak négy szem-elv szerinti rögzítést, a kritikus jellemzők, rendszerparaméterek módosítása során.

Megvalósítási lehetőségek

Az üzleti alkalmazások működését befolyásoló paraméterek módosítását általában egy felhasználói felületen keresztül tudja elvégezni a megfelelő jogosultságokkal rendelkező üzleti szereplő. Hasonló módon, a tömeges adatbetöltéseket és adatmódosításokat speciális programokkal lehet megvalósítani. A nemzetközi háttérű, dobozos üzleti szoftvergyártók (pl. SAP, SAS, Microsoft, Oracle stb.) termékpalettáján általában rendelkezésre állnak a saját megoldásukhoz kifejlesztett, üzleti paraméterszerkesztést vagy tömeges adatműveleteket lehetővé tevő interfész alkalmazások.

Amennyiben ilyen eszköz nem áll rendelkezésre, lényegében három különböző alternatíva létezik, a rendszer-paraméterezési-, adatbetöltési- és tömeges adatmódosító feladatok végrehajtásához:

- **közvetlen adatmódosítás:** a produktív rendszer megfelelő táblájában történő módosítás vagy betöltés,
- egyedi fejlesztésű, **adatmódosító patchprogramok** használatával,
- **általános célú alkalmazás** segítségével.

A következőkben fenti alternatívákat vizsgáljuk meg részletesebben.

Közvetlen adatmódosítás

Előfordulhatnak olyan rendkívüli helyzetek, amikor a leggyorsabb vagy leghatékonyabb megoldásnak az tűnik, hogy a rendszergazda támogatásával közvetlenül az éles adatbázisban módosítanak egy-egy paraméterértéket, vagy egy SQL script befuttatásával módosítanak adatokat. Ennek a megoldásnak a legnagyobb kockázata, hogy az adatmanipuláció az alkalmazás által biztosított kontroll funkciók megkerülésével történik, ezáltal az üzembiztonságot közvetlenül veszélyeztetheti. Ezesetben a végrehajtott műveletek nem rekonstruálhatók utólag, nem vonható vissza, és nem tesztelhető az élesítés előtt a módosítás helyessége, így az adatkonzisztencia is sérülhet.

- **Előnyök:**

- *gyorsan elvégezhető a módosítás,*
- *nincs szükség rendszerfejlesztésre,*
- *a javítás eredménye azonnal ellenőrizhető.*

- **Hátrányok:**

- *sérülhet az adatkonzisztencia,*
- *körütekintő kockázatfeltárást követően lehet alkalmazni,*
- *az elvégzett művelet nem feltétlenül vonható vissza.*

Adatmódosító patchprogram

Ennél az alternatívánál az adatkarbantartás végrehajtása már standardizált módon, a „normál” rendszerfejlesztési folyamathoz hasonlóan történik azzal a különbséggel, hogy ez esetben csak az adattartalom változik (datafix). A patchprogram futtatásával fenntartható a konzisztens rendszerállapot, és utólag is visszakövethetők a végrehajtott műveletek, teljesíthetők az audit követelmények. Az összeállított programcsomag újrafelhasználható, vagyis az adatok cseréjével (pl. külső forrásból származó állomány feltöltése adatbázisba) az alap eljárás többször is használható.

Ugyanakkor, a külső forrásból származó adatok adatbázisba töltését, vagy a tömeges adatmódosítást megvalósító patchprogramok tervezése, telepítése és futtatása gyakran sokszereplős, hosszú átfutási idővel járó feladat. Ez a megoldás meglehetősen „IT-intenzív”, mivel a megoldáshoz számos szakértő összehangolt közreműködésére van szükség.

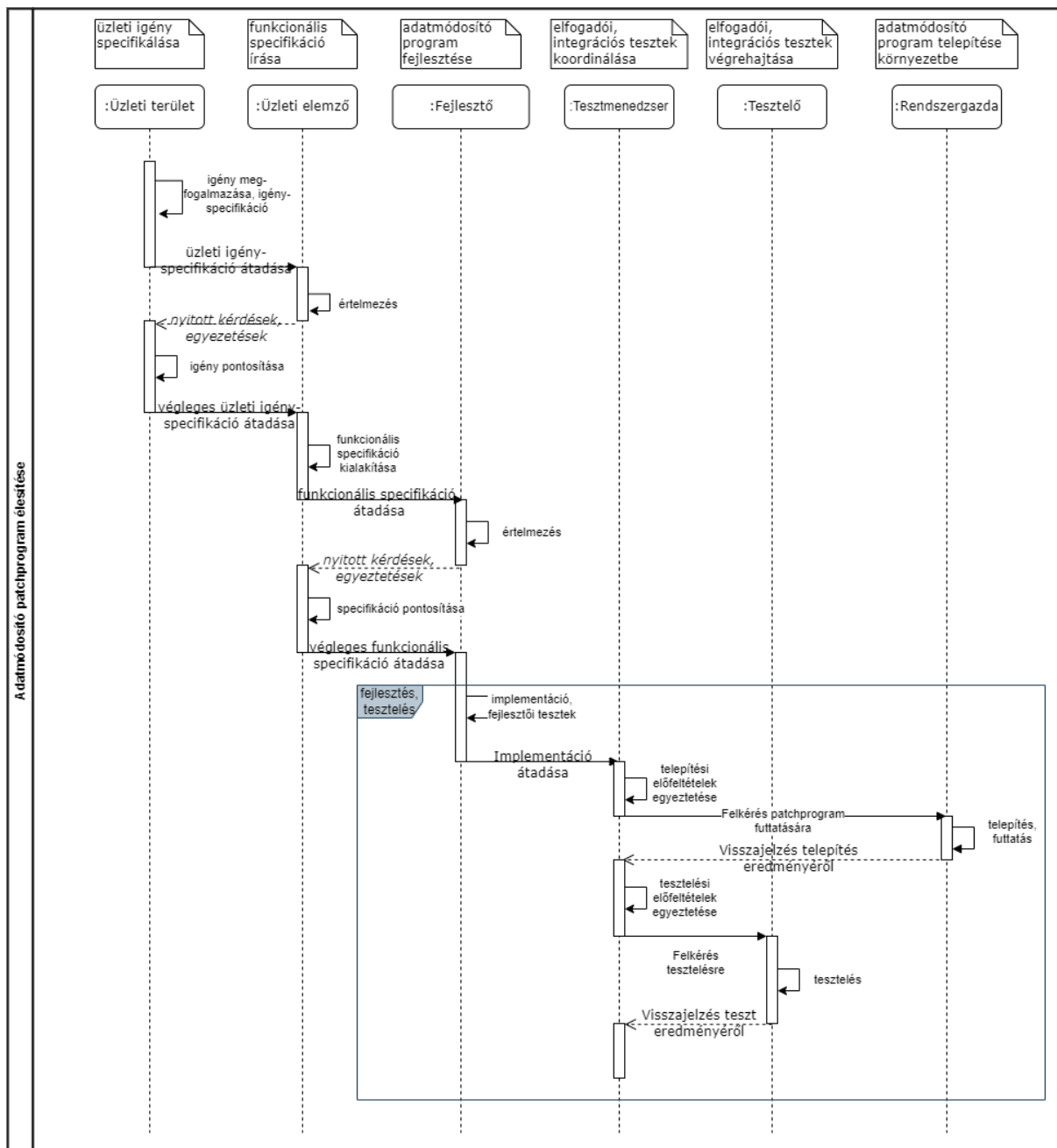
- **Előnyök:**

- *üzembiztos, standardizálható megoldás,*
- *verziókezelhető, újrafelhasználható kód jön létre,*
- *a módosítások végrehajtása során adatbázis szintű validációs eljárások alkalmazhatók.*

- **Hátrányok:**

- *hosszabb átfutási idő,*
- *egy datafix eljárás csak egy célra használható – végrehajtandó műveletenként különböző eljárás implementálása szükséges,*
- *az elkészült fejlesztést körültekintően tesztelni kell élesítés előtt.*

A következő ábra egy adatkód módosító patchprogram fejlesztésének egy lehetséges szekvenciadiagramját mutatja be, amelyen láthatók a fejlesztés élesítésében közreműködő szereplők, feladataik, illetve a közöttük zajló interakciók.



1. ábra Patchfolyamat szekvencia diagramja

Az ábrán szereplő aktorok és a patchmenedzsment folyamatban betöltött szerepük:

- **üzleti terület** (szakterület): a fejlesztési igényt elindító szereplő, akinek célja valamely üzleti folyamat hatékonyságának növelése
 - o *bemeneti kritérium*: üzlet haszonszerzés (business case)
 - o *kimeneti kritérium*: üzleti igény specifikációs (ÜIS) dokumentum, amelyben meghatározásra kerülnek a fejlesztéssel kapcsolatos felsőszintű üzleti követelmények

- **üzleti elemző** (BA, rendszerszervező, folyamatszervező): az üzleti igény alapján, specifikálja a célrendszerben megvalósítandó fejlesztést, definiálja az elvárt funkcionális és nem funkcionális követelményeket
 - o *bemeneti kritérium*: jóváhagyott üzleti igény specifikációs dokumentum
 - o *kimeneti kritérium*: funkcionális specifikáció
- **fejlesztő**: adott funkció / alkalmazás / szolgáltatás / rendszerkomponens implementációjáért felelős szereplő, aki az adott fejlesztést leszállítja
 - o *bemeneti kritérium*: jóváhagyott funkcionális specifikáció
 - o *kimeneti kritérium*: telepíthető alkalmazás / szolgáltatás / rendszerkomponens
- **tesztmenedzser** (környezetmenedzser): az átadott fejlesztések telepítési- és tesztelési előfeltételeinek biztosításáért felelős, elfogadói-, integrációs- és éles környezetekben
 - o *bemeneti kritérium*: formai követelményeknek megfelelő rendszermódosító (verzióváltó) programcsomag
 - o *kimeneti kritérium*: programcsomag telepítésének menedzselése teszt-, integrációs- és éles környezetben, valamint tesztelés végrehajtásának koordinálása
- **tesztelő** (környezetmenedzser): az elfogadói-, integrációs- és éles környezetekbe telepített fejlesztés elvárásoknak megfelelő működésének ellenőrzése, a funkcionális specifikációban leírt követelmények alapján
 - o *bemeneti kritérium*: telepített alkalmazás / szolgáltatás / rendszerkomponens
 - o *kimeneti kritérium*: tesztjegyzőkönyv, hibajegyek
- **rendszergazda** (adatbázis adminisztrátor, DBA): az átadott fejlesztés elfogadói-, integrációs- és éles környezetekbe való telepítését végzi
 - o *bemeneti kritérium*: formai követelményeknek megfelelő rendszermódosító (verzióváltó) programcsomag
 - o *kimeneti kritérium*: telepítési jegyzőkönyv, hibajegyek

Megjegyzés: az elkészült fejlesztés tesztelése, az esetleges hibajavításokkal együtt általában több iterációs ciklusban valósul meg, többféle (pl. elfogadói-, integrációs, pre-prod, stb.) környezetben.

Általános célú alkalmazás

Az általános célú alkalmazások legnagyobb előnye, hogy az adatkarbantartási művelethez egy felhasználói felület áll rendelkezésre, amely segítségével az üzleti felhasználók az IT szakértők bevonása nélkül tudják elvégezni az éles rendszerben a kívánt módosításokat. A végrehajtott változások véglegesítése előtt lehetőség van validációs szabályok futtatására, jóváhagyási workflow kialakítására, amelyekkel megelőzhető a téves rögzítésből adódó hibázás lehetősége.

Az adattárház folyamatok támogatásához számos open source és kereskedelmi termék áll rendelkezésre:

- a külső forrásból származó adatállományok (pl. csv, xlsx) adatbázisba töltéséhez a különböző ETL eszközök (pl. Informatica PowerCenter, Apache Airflow, IBM Infosphere Datastage, Oracle Data Integrator, Talend Open Studio, Pentaho Data Integration, stb.) nyújtanak hatékony támogatást,
- a tömeges adatmódosításhoz vagy adatkorrekciók futtatásához általában az adatelőkészítésre, adattisztításra szánt alkalmazások (pl. IBM InfoSphere QualityStage, Informatica Cloud Data Quality, SAS Data Quality, Data Ladder, OpenRefine, WinPure, stb.) használhatók,
- az üzleti paraméterek, forgatókönyvek felhasználói felületen való karbantartásához pedig a különböző „low-code” platformok (pl. Oracle Apex, DBeaver, Microsoft Power Apps, OutSystems, Mendix, Kissflow, stb.) biztosítanak lehetőséget.

Fenti alkalmazásokból összeállítható egy olyan szakértői környezet, amelyekkel kielégíthetők az üzleti felhasználók igényei, azonban érdemes megjegyezni, hogy az egyedi fejlesztésű célalkalmazások fenti funkciókat képesek egyben megvalósítani.

- **Előnyök:**

- o *robotizált, hibátűrő megoldás hozható létre,*
- o *az üzleti felhasználók az IT terület bevonása nélkül is el tudják végezni a szükséges módosításokat,*
- o *lehetőség van alkalmazás által nem biztosított kontrollpontok beépítésére, kiegészítésére.*

- **Hátrányok:**

- o *új alkalmazás bevezetésére lehet szükség, ezáltal költségesebb alternatíva,*
- o *a társrendszerekkel való integráció előfeltételeinek megteremtése hosszabb tervezést igényel,*
- o *az általános célú alkalmazások nem mindig optimálisak egy adott feladathoz vagy használati esethez.*

Összefoglalás

Az alábbi táblázatban az éles üzemben végrehajtandó adatkarbantartó, adatmódosító műveletekkel kapcsolatos legfontosabb szempontokat foglaltuk össze, megjelölve, hogy a fenti alternatívák közül melyek tudják teljesíteni az adott elvárást.

Szempont/elvárás	Közvetlen adatmódosítás	Adatmódosító patchprogram	Általános célú alkalmazás
Auditálhatóság: a végrehajtott módosítások utólag is rekonstruálhatók?	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
Validáció: van-e lehetőség a módosítások végrehajtása előtt az adatok szintaktikai / szemantikai ellenőrzésére?	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
Üzembiztonság: a módosítások végrehajtását követően fenntartható a konzisztens rendszerállapot?	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
Folyamati kontroll: van-e lehetőség célalkalmazásban nem implementált kontrollfolyamat utólagos beépítésére?	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Újrafelhasználhatóság: az eljárás későbbi időpontban is használható azonos célra?	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
Rugalmasság: pontosan illeszthető az adott rendszerhez?	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Teljesítmény: adatmennyiség növekedése esetén sem romlik számottevően a performancia?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Megvalósítás időszükséglete: rövid átfutási idővel megvalósítható az adatmódosítás?	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

ClarA adatkarbantartó és -menedzsment megoldás

Jogosan merül fel a kérdés, hogy miért tartottuk fontosnak a problémakör és megvalósítási lehetőségek részletes bemutatását. Ennek magyarázata az, hogy az elmúlt évtizedek során számos esetben talákoztunk Ügyfeleinknél a bemutatott kérdéskörrel. Azt is tapasztaltuk, hogy korántsem triviális a felmerülő nehézségekre rövid átfutási idővel bevezethető, gazdaságos, ugyanakkor hosszú távon fenntartható megoldást találni.

A probléma aktualitását felismerve, kollégáink egy olyan alkalmazás tervezésén kezdtek el dolgozni, mely jelentős mértékben megkönnyítheti egy adattárház töltési folyamatainak menedzselését. A tervezés során a legfontosabb alapelvünk az volt, hogy az egyes karbantartási feladatokat az a munkatárs végezze el, aki egyrészt birtokában van a megfelelő információknak, másrészt felelősséggel tartozik az adott művelet végrehajtásáért.

Az alkalmazás fejlesztési életciklusa egészen 2007-ig nyúlik vissza, amikor az általunk szállított megoldással Ügyfelünk számára lehetővé tettük, hogy az adattárház dimenzió tábláit, manuális input állományait, valamint üzleti szűrőfeltételeit az üzleti felhasználók egy felületen keresztül tarthassák karban.

Az elmúlt 17 évben folyamatosan fejlesztettük termékünket, és számos refaktorálást követően, mostanra értünk el oda, hogy a ClarA termékünk egy olyan dobozos megoldás legyen, mely stabil, könnyen bevezethető, ergonomikus megoldást nyújt, a cikkünk első részében tárgyalt problémákra.

Megvalósítását tekintve, mely csoportba sorolnánk a ClarA-t? Egyértelműen nem kategorizálható be egyik csoportba sem, hiszen a közvetlen adatmódosításnál és adatmódosító patchprogramoknál sokkal fejlettebb, hibatűrőbb megoldást nyújt. Ugyanakkor, általános célú alkalmazásnak sem tekinthető, mert sokkal fókuszáltabban kezeli az adattöltések menedzselését.

A rendszer bevezetése óta felmerült üzemeltetési tapasztalataink és a felhasználói visszajelzések alapján, sikerült egy olyan általános célú, moduláris felépítésű webes alkalmazást létrehoznunk, amely az Ügyfél rendszereihez integrálódva, lehetővé teszi a napi működéshez szükséges paraméterezési- és karbantartási feladatok kontrollált végrehajtását. Az alkalmazás alapvető célja, hogy egységes rendszerben kezelje mindazon fentebb bemutatott alkalmazásüzemeltetési-, adminisztratív- vagy üzleti jellegű paraméterezési igényeket, folyamatokat, amelyek nagyvállalati környezetben felmerülhetnek.

A rendszer elsődleges felhasználói általában az üzleti elemzők, adatelemzők, illetve az üzleti intelligenciával foglalkozó szakemberek.

A ClarA használatával elérhető előnyök:

- Használatával **felgyorsítható** a patchfolyamat átfutási ideje azáltal, hogy az üzleti területek (auditált körülmények között), közvetlenül tudnak az éles rendszerben adatokat módosítani, fejlesztés nélkül.
- **Moduláris felépítésének** köszönhetően, az egyes modulokban megvalósított üzleti funkciók függetlenek a többi modul meglététől, és azok használatától.
- Az **üzleti funkciók zártsága** biztosítja az operációs kockázatok csökkenését, egyúttal hozzájárul az adatminőség javulásához.
- Korszerű, **felhasználóbarát felhasználói felülettel** rendelkezik, amelynek használata könnyen elsajátítható.

A ClarA által nyújtott funkciók, szolgáltatások leggyakoribb alkalmazási területei:

- külső forrásból származó adatállomány feltöltése adatbázisba,

ClarA Demo környezet Felhasználó Ferenc

Környezetek > Modulok > Táblázatfeltöltések > Csoportok > CLA_DEMO01 csoport > CUP_DEMO_24 feltöltés > Fájlfeltöltés

Fájlfeltöltés CUP_DEMO_24 Tábla

Véglegesítés (0) Mégsem

FÁJL clara_upload_sheet#1-minta_cup_demo_24_ab-minta_cup_demo_24_ab

Azonosító Fájl

22 clara_upload_sheet#1-minta_cup_demo_24_ab-minta_cup_demo_24_ab.csv Fájlfeltöltés

Leírás

Karakteres adat Bináris adat

Feltöltés típus név Feltöltés típus megnevezés

CUP_WORKBOOK CUP_WORKBOOK

Verzió Utolsó módosító Utolsó módosítás

1 felhasznalo.ferenc 2023.10.06 12:28:56

- üzleti és/vagy technikai paraméterek karbantartása,
- értéklistán, értékkeszletek, dimenziótáblák karbantartása,

ClarA Demo környezet Felhasználó Ferenc

Környezetek > Modulok > Forgató-és Paramétertáblák > Csoportok > CLA_DEMO301 csoport > CMAP_DEMO_32 tábla > CMAP_DEMO_32 info

CLARA_DEMO.CMAP_DEMO_32 infó

JELLEMZŐK OSZLOPOK **FELSOROLÁSOK** HIBAÜZENETEK

Oszlop név

OSZLOP NÉV	ÉRTÉK	MEGNEVEZÉS	INDEX
FLAG	YES	Yes	1
FLAG	NO	No	2

- adatok tömeges módosítása, auditált korrekciójása.

ClarA Demo környezet Felhasználó Ferenc

Környezetek > Modulok > Módosító Szkriptek > Futási naplók > 2023-09-25 > CMOD_DEMO_10 > #1 futás

CMOD_DEMO_10 2023-09-25 #1 futás

SZKRIPT FUTÁS SZKRIPT NAPLÓ EREDMÉNY NAPLÓ JOB FUTÁS JOB NAPLÓ

Gyöker futás Szülő futás

1

Szkript név Megnevezés Dátum

CMOD_DEMO_10 Clarity Demo-1 2023.09.25

Szkript csoport Megnevezés Partíció lista Partíció megnevezés

CLA_DEMO301 Demo script group-1 DATE Dátum

Azonosító Verzió Utolsó módosító Utolsó módosítás

1 0 olah.beatrix 2023.09.25 14:45:14

Szkript

```

DECLARE
  l_run_id      INTEGER;
  l_run_data   clara_run_data_tp;
  l_xresult    clara_run_result_tp;

BEGIN
  -- Setting log level. If not set, it inherits from the application.
  -- Naplózási szint beállítása. Ha ez nem történik meg, akkor örökli az alkalmazástól.
  clara_script_if_pck.set_log_level(clara_script_if_pck.c_log_level_debug);

  -- Current run identifier
  -- Aktuális futás azonosítója
  l_run_id := clara_script_if_pck.get_run_id;

  -- Current run data
  -- Aktuális futás adatai
  l_run_data := clara_script_if_pck.get_run_data(l_run_id);

  -- Writing run log
  
```

További kérdése merült fel az adattárhízi folyamatok adatmenedzsmnt folyamataival kapcsolatban, esetleg Önt is foglalkoztatja a paraméterezési-, üzemeltetési feladatok költségkhatékony ám biztonságos

megoldásának kérdése? Tanácsadói és fejlesztői csapatunk szívesen áll a rendelkezésére, a következő elérhetőségek valamelyikén:

Clarity Consulting Kft.

+36 1 422 3030

info@clarity.hu