

K2

Gyakori kérdések

Készítette: Informatikai Fejlesztési Igazgatóság
Verzió: 1.2
Dátum: 2011. január 17.
Elhelyezés:
Fájlnév: Rendszerterv sablon.docx

Tartalom

Bevezetés	3
Telepítés	3
Éles környezet	3
Tesztkörnyezet	3
Könyvtárszerkezet	4
Memória táblák és működés	4
Üzemeltetés	5
Konfigurálás	7
Az IFSC API programozása	8
Ajánlat beküldés	10
Ajánlat módosítás (törlés+új)	10
Ajánlat visszavonás	11
Árjegyzői ajánlatpár	11
Ajánlatok lekérdezése	11
Kötések lekérdezése	11
Ajánlati könyvek figyelése	12
Egyéb kérdések	12

Módosítások jegyzéke

Verzió	Szerző	Dátum	Módosítások összefoglalása
1.0	Pollák A.	2011.01.11	Első verzió
1.2	Vass Attila	2011.01.17	Pontosítások

Bevezetés

Jelen dokumentum azzal a céllal készült, hogy a K2 szoftvert használó szekciótágjainkat támogassuk a K2 mindennapi működésével, beállításával és az üzemeltetés során felmerülő kérdésekkel kapcsolatban. (A leírás megértéséhez szükség van a K2 dokumentációk ismeretanyagára.)

A K2 segítségével a BÉT mindkét – azonnali és derivatív piaci - rendszeréhez is lehet kapcsolódni. MMTS 1 –nek rendszernek hívjuk az azonnali piacot működtető rendszert, és MMTS 2 –nek a derivatív piacot működtető rendszert.

A legtöbb esetben a K2 programkomponensei is szétválnak 1-es és 2-es rendszerre. A dokumentum további részében a példákban az 1-es rendszert megcélzó parancsokkal foglalkozunk, vagyis a parancs nevében, a parancs paraméterében, vagy a paraméterfájl nevében az 1-es szám fog szerepelni, de ezek a 2-es rendszerre is ugyanúgy működnek.

Telepítés

Éles környezet

■ **Hogyan történik a K2 első telepítése?**

A **K2_inst_guide_1_7H.pdf** dokumentáció utasításai szerint kell eljárni. Ez a dokumentáció megtalálható a BÉT honlapján itt:

http://www.bet.hu/topmenu/tozsdei_kereskedok/members_techikai_informaciok/members_techikai_dokumentacio/members_kerszoft_doksi.html

K2 dokumentációk összecsomagolva tölthetők le a **k2doc.zip** néven.

E témához kapcsolódva érdemes elolvasni az Egyéb kérdések fejezetben a K2 és a TW szoftver összehasonlítását tárgyaló részt.

■ **Telepítés után hiányoznak a konfigurációs állományok a k2/cfg könyvtárból. Ezeket hogyan pótolhatom?**

Fenti könyvtár tartalmát (5 db fájlt) utólag küldjük el, amikor a felhasználó szekciótág vagy fejlesztő cég már aláírta a K2 szerződést.

■ **Verziófrissítés esetén mit kell tennem?**

Rendelkezésre áll egy rövidebb dokumentum a fent említett web oldalon „K2 telepítése” címszó alatt **K2_telepitesi_utmutato_2007_06.pdf** néven melynek utasításait kell követni.

Tesztkörnyezet

■ **Telepíthetem a tesztkörnyezetemet ugyanarra a gépre, ahol az éles is működik?**

Tapasztalataink szerint az éles és a teszt környezet egy hardveren is tud működni, ha egyik sincs extrém módon terhelve. Az éles környezet egy **k2** nevű Linux/Unix felhasználó alatt kerül telepítésre, míg a teszt környezet egy **testk2** felhasználó alatt.

További eltérő beállítások:

- **test1** (és test2) szerviz (port) felvétele szükséges az „/etc/service” fájlban (lásd **K2_inst_guide_1_7H.pdf** dokumentáció 17. oldal)
- **k2/cfg/pgw1.cfg** konfigurációs fájlban a következő paraméterek beállítása kell eltérjen:
 - ipckey* (a memóriatáblákra hivatkozó kulcsszám)
 - ifssservice* (élesben: **ifss**, illetve **ifss2**; teszten: **test1** és **test2** az értéke)

- logpath* (csak akkor van különbség, ha teljes útvonal van megadva; pl /home/k2/k2/log)
- rlogpath* (ugyanaz a feltétel, mint az előbbi logpath esetében)

- a **k2/cfg/pgwtsmr1.ini** konfigurációs fájlban a **broadcastAddress** paraméter a BÉT teszt szerverére mutató IP cím legyen:

broadcastAddress=192.168.15.1

- **Milyen előnyei vannak, ha külön gépre telepítem a tesztkörnyezetet?**

Ha a tesztkörnyezet egy külön gépre kerül, akkor szinte minden beállítás megegyezhet az élesrel. Az egyetlen kivétel az előzőekben említett IP cím (**k2/cfg/pgwtsmr1.ini** konfigurációs fájl, **broadcastAddress** paraméter). Így ez a környezet egyben tartaléka is lehet az élesnek.

Könyvtárszerkezet

- **Mit és hol találok a telepítés után a kicsomagolt könyvtárszerkezetben?**

- **k2/bin** programok és parancsfájlok helye.
- **k2/cfg** konfigurációs fájlok helye (ifss.uaf, pgw1.cfg, pgw2.cfg, pgwtsmr1.ini, pgwtsmr2.ini).
- **k2/log** naplófájlok helye (pgw1.log, ifss1.log, pgw2.log, ifss2.log); továbbá ide kerülnek az esetleges újraindításhoz szükséges napi tranzakciós fájlok (restart log) (YYYYMmmDD_X.rlg , ahol X 1 vagy 2).
- **k2/pkg** API programozásához szükséges fájlok (és példaprogramok) helye, mely további alkönyvtárakat tartalmaz.

Memória táblák és működés

- **Milyen processzek dolgoznak a memóriatáblákkal és milyen szerepük van?**

A **pgw1** processz (programkomponens) tartja a kapcsolatot a BÉT kereskedési rendszerével és az onnan lekérdezett adatokat memóriatáblákban tárolja. Az **ifss1** processz szolgálja ki a K2 kliens programokat, melyhez az adatokat a pgw1 processz által írt memóriatáblákból veszi. Az **ifss1** fogadja a tranzakciókat a kliensektől és tárolja a megfelelő memóriatáblában (orderentry, mmorder), amelyeket aztán a **pgw1** továbbküld a kereskedési rendszerbe.

A memóriatáblák részletes leírása a **K2_refifsc_2_9H.pdf** dokumentációban olvasható a 34. (és 35.) fejezet alatt.

- **Milyen memóriatáblái vannak a K2-nek?**

market, instr, sector, board, secboard, firm, user, order, trade, negdeal, orderentry, mmorder, auditevent, priceparam, systime.

- A **systime** egy speciális tábla, csak egy információs rekordból (sorból) áll. API oldalról ennek a táblának a lekérdezése is különbözik a többitől (más függvényvel oldható meg). Alkalmas a **pgw1** processz monitorozására is API oldalról.
- A **secboard** és a **priceparam** táblák pontosan ugyanannyi sorból állnak. Tulajdonképpen a priceparam kiegészítő táblája a secboard-nak.
- Az **orderentry** és **mmorder** táblák tartalmazzák a kliensek által beküldött tranzakciókat, vagyis az **ifss1** processz írja a rekordjait. A **pgw1** processz a kereskedési rendszertől kapott válasz alapján e tábláknak csak néhány mezőjét frissíti, amelyek közül a legfontosabb a státusz mező.

Üzemeltetés

■ **Futhat-e folyamatosan 24 órában a K2?**

A K2 rendszert naponta kell indítani és leállítani. Indítására akkor van lehetőség, ha a BÉT kereskedési rendszere már működik és engedi a felhasználók bejelentkezését (8:00-kor a legtöbb esetben ez sikerülni fog). A kereskedés lezárását követően (utolsó kereskedési szakasz vége után) még legalább 20 percig lehetnek bejelentkezve a felhasználók (jelenleg 17:30-ig). Ekkor a BÉT üzemeltetése kijelentkezeti a felhasználókat, ami K2 oldalon a pgw1 processz leállításához vezet, de ettől még az ifss1 processz üzemképes marad, és tudja fogadni a K2 kliensek kéréseit, valamint a napi kereskedési adatokat a memóriatáblákból szolgáltatni tudja, de tranzakciót már nem érdemes beküldeni, mert azok már nem fognak végrehajtódni. A K2 leállítását ezek után bármikor megtehetjük.

■ **Lehet-e automatikusan indítani és leállítani a K2-t?**

Linux / Unix környezetben a **crontab** segítségével ütemezetten tudunk parancsokat végrehajtatni. Így lehetőség van a K2 rendszer indítására és leállítására kézi beavatkozás nélkül is. A k2start parancsnak -p opcióval lehet megadni a belépéshez szükséges jelszót, amit kézi indításnál a billentyűzetről vár tőlünk.

■ **Hogyan tudok a K2-vel jelszót módosítani?**

A BÉT kereskedési rendszerébe való első bejelentkezéskor, az összes jogosult funkció eléréséhez, szükséges a BÉT által kiadott kezdeti jelszót megváltoztatni. Jelszócseréhez a **k2changepwd** parancsot használhatjuk még a k2start kiadása előtt!

■ **Hogyan tudom a processzeket és a memóriatáblákat monitorozni?**

A **mem1** parancs által kiírt lista információt nyújt mind a processzekről, mind a memóriatáblákról:

```
==> k2test2 1
KEY: 0x70677731 SHMID: 2785282 SEMID: 3244032
-----
          Num      Max      Recsize
market          4         4         48
sector         31        31         44
instr           44        44         48
board           25        25         88
secboard       552       552       1392
firm           623       623         300
user           355       356         324
order          375    30000         248
trade           45   120000         212
negdeal         0     1000         172
auditevent_t    48     500          60
auditevent_g     9     500          56
auditevent       3   60000         332
orderentry     208   50000         232
mmorder         0    10000         300
badreplymsg     4    25000         256
-----
Last orderno:          588061
Last tradeno:          178879
Last last_n tradeno:   178879
Last negdealno:       588061
```

```

Stats Version:      0x4d2a9240
Log Time:           0
Transactions:       64
Rejections:        0
  
```

```

-----
      pid      ppid comm  vshm   shm  starttime  utime  stime status
00027396 00000001 pgw1   n/a   n/a   4057832    0    0 Sleep
00027399 00000001 ifss   n/a   n/a   4057837    0    0 Sleep
00027401 00027396 pgw1   n/a   n/a   4057843    0    0 Sleep
00027714 00027687 mem1   n/a   n/a   4078363    0    0 Run
00027715 00027714 mem1   n/a   n/a   4078363    0    0 Run
=====
  
```

A lista első felében lehet nyomon követni a memóriatáblák telítettségét. A tábla megnevezése mellett három szám szerepel, melyből az első kettő a fontos, a felhasznált rekordok (Num), és az összes felhasználható rekord száma (Max). Ha az első szám eléri a másodikat, akkor a műveletet végrehajtó processz hibával leáll. (A **pgw1** leáll az *order*, *auditevent*, *badreplymsg* táblák telítődése esetén; **ifss1** leáll az *orderentry*, *mmorder* telítődése esetén).

A középső rész kevésbé fontos. Az utolsó részben a K2-höz tartozó futó processzek listája található.

Ha csak a processzeket szeretnénk megvizsgálni, akkor például a

```
ps -fu k2 | grep pgw1
```

parancs kilistázza k2 felhasználó pgw1 processzeit, melyből normál üzemmód esetén két példány fut (egyik tartja a kapcsolatot a kereskedési rendszerrel, a másik meg figyeli az egyik működését). A

```
ps -fu k2 | grep ifss1
```

parancs kilistázza a k2 felhasználó ifss1 processzeit. Normál esetben 1+n darab példány fut, ahol az „1” fogadja a kliensek csatlakozási kéréseit, a „n” pedig a már létrejött klienskapcsolatok száma. Minden klienskapcsolatot külön ifss1 processz szolgál ki.

A memóriatáblák tartalmát kilistázhatjuk a **get_table** parancs segítségével, mely parancs kliensként jelentkezik be a rendszerbe, és úgy hajtja végre a lekérdezést.

A **dump1** parancs ömlesztve írja ki a memóriatáblák tartalmát, mely nehezebben olvasható, de jól jöhet arra az esetre, ha az ifss1 processzek mind elhaltak, és nincs lehetőség kliens bejelentkezésre.

■ **A naplófájlokban hogyan kereshetek hibaüzeneteket?**

A **pgw1** processz írja a *k2/log/pgw1.log* naplófájlt, az **ifss1** processz írja a *k2/log/ifss1.log* naplófájlt stb. Minden naplóbejegyzés (sor) egy azonosító résszel kezdődik, amit az információs rész követ. Az azonosító rész végén szerepel egy kötőjel után egy betűjel, ami **I**=információs sor; **W**=figyelmeztető üzenet; **E**=hibaüzenet lehet. Ha tehát hibaüzenetet akarunk keresni, akkor ezekre az „E” betűs sorokra kell rákeresni, például így:

```
grep [-]E ifss1.log
```

Ezek a naplófájlok, ha nem távolítjuk el közben, akkor napról napra folyamatosan bővülnek, így tartalmuk egyre átláthatatlanabb lesz. Figyelni kell arra is, hogy ha a mérete eléri a 2 GB -ot , akkor a processz nem tudja tovább írni, és hibaüzenettel leáll. Ezért meghatározott időközönként gondoskodni kell e naplófájlok mentéséről és eltávolításáról (akár naponta a zárás után).

■ **Hogyan nézhetem meg utólag a K2-be beküldött tranzakciókat?**

A *k2/log* könyvtárba készülnek a tranzakciós naplófájlok is. Naponta és rendszerenként (1-es és 2-es) egy-egy fájl. Ezek tartalma csak segédprogram segítségével olvasható. Például, ha ki akarom listázni a 2011. január 3-i állományt, akkor a következő paranccsal tehetem meg:

```
rldump1 2011Jan03_1.rlg
```

■ **Mikor szükséges napközben újraindítani a K2 rendszert, és azt hogyan tehetem meg?**

Előfordulhat, hogy hálózati probléma miatt megszakad a kapcsolat a pgw1 processz és a kereskedési rendszer között, ami a pgw1 leállításához vezet. Továbbá előállhat olyan eset, hogy valamely memóriatábla betelik, ami szintén valamelyik processz leállításához vezet. Ilyenkor szükség van újraindításra, ami egy leállítás paranccsal kezdődik (**k2stop 1**). Ha valamelyik memóriatábla betelt, akkor gondoskodni kell a megfelelő tábla átméretezéséről a *k2/cfg/pgw1.cfg* állományban. Ezek után normál módon indítható a rendszer (**k2start 1**). Indulás után az orderentry és mmorder táblák is feltöltődnek a tranzakciós naplófájl beolvasásakor, melynek során a leállítás előtti állapotot kapjuk vissza. Fontos, hogy az újraindítás után a változásszám (seqno) mező értékek újra generálódnak, tehát a K2 kliens ilyenkor a leállítás előtti változásszámokra nem építhet!

Konfigurálás

■ **Mely paramétereket állíthatom át a pgw1.cfg konfigurációs állományban?**

A módosítható paraméterek: *ipckey, ifssservice, logpath, rlogpath, verbose, foreground, watchpgwenabled, freesemenabled, maxorderentry, maxbadreplymsg, maxmmorder, maxauditeventtype, maxauditevent, firstorderentry, numoforderbook, orderratio, password, skiplogchk, tsmrtimeout, timesyncntimeout, timesyncnretry, timesyncncycle, ordlistsize* (ez utóbbi csak a pgw2.cfg –ben lehet)

A legtöbb paraméter beállításán nem szükséges változtatni. Ha nincs is beállítva a konfigurációs fájlban, akkor az alapértelmezett értéke van érvényben.

Paraméterek, amelyeket előfordulhat, hogy szükséges átállítani:

- Élessel egy gépen futó tesztkörnyezetben más értéket kapnak:
 - ipckey, ifssservice, logpath, rlogpath*
- Memóriatáblák méretezéséhez szükséges paraméterek:
 - maxorderentry* az orderentry tábla méretét adhatjuk meg ezzel; több kell legyen mint a tervezett napi tranzakciók száma.
 - maxbadreplymsg* a badreplymsg tábla méretét adhatjuk meg ezzel; ebben tárolja a K2 a hibás (elutasított) tranzakciók hibaüzeneteit.
 - maxmmorder* a mmorder tábla méretét adhatjuk meg ezzel; több kell legyen mint a tervezett napi árjegyzői ajánlatpár-tranzakciók száma.
 - numoforderbook* ezzel a paraméterrel határozhatjuk meg, hogy legfeljebb hány darab ajánlati könyvet szeretnénk figyelni; értéke lehet nagyobb az orderbookcyclic értékénél.
 - orderratio* az order tábla méretét szabályozhatjuk ezzel a paraméterrel; a kereskedési rendszerben lévő order tábla méretét osztja ezzel a paraméterrel és úgy méretezi a K2 order tábláját; alapértelmezett értéke: 10; normál esetben ez a tizednyi táblaméret elegendő, mivel a cég csak a saját ajánlatairól kap információt.

■ **Melyek a licenc védett paraméterek?**

A licenc védett paraméterek: *username, firmname, cycletime, orderentrypercycle, timesyncenabled, intrefenabled, orderbookcyclic, sleepimit, sendsignal, cyclicobenabled, fieldchgenabled, retrieveordno, marketbyprice, marketbyorder, orderbookchange, numofclients, license*.

A licenc védett paramétereket nem szabad módosítani!

■ **Mi a különbség licenc típusok között?**

Három licenc típusból lehet választani: ENTRY, STANDARD, PLUS, melyek néhány licenc védett paraméter értékében térnek el egymástól:

	ENTRY	STANDARD	PLUS	
orderbookcyclic	0	50	100	egy programciklusban lekérdezhető ajánlati könyvek száma
cyclicobenabled	0	1	1	engedélyezi a ciklikus ajánlati könyv lekérdezést
marketbyprice	0	1	1	árszint összesített ajánlati könyveket lekérdezhet
marketbyorder	0	0	1	ajánlat szintű ajánlati könyveket lekérdezhet
sendsignal	0	1	1	új tranzakció esetén ébressze fel a pgw1 processzt
fieldchgenabled	0	1	1	secboard tábla dinamikus mezőire engedélyezi a mezőszintű lekérdezést
retrieveordno	0	1	1	az új befogadott ajánlat számát írja be az orderentry tábla OrdNo mezőjébe
numofclients	2	10	10	legfeljebb ennyi ifss1 processzt enged párhuzamosan futni; kliens kapcsolatok számának korlátozása

■ **Milyen jogosultságai lehetnek egy K2 kliensnek, és ezt hol tudom állítani?**

A **k2/cfg/ifss.uaf** fájlban vannak leírva a K2 kliensfelhasználók jogosultságai. Ez egy szövegszerkesztővel módosítható fájl. Egy-egy sora négy darab kettősponttal elválasztott mezőből áll: username:password:active/suspended:vesszővel elválasztott jogosultságok

Példák:

broker1:pwd1:s:query	felfüggesztett felhasználó – nem tud belépni
broker2:pwd2:a:query	csak lekérdezési jogosultsággal rendelkező felh.
broker3:pwd3:a:query,entry	ajánlatot beküldhet, de az csak „A” accepted – elfogadott státuszt kap; jóváhagyása szükséges
broker4:pwd4:a:query,entry,confirm	jóváhagyási jogosultsággal rendelkező felhasználó
broker5:pwd5:a:query,entry,confirm,bypass	a beküldött ajánlata egyből „C” confirmed – jóváhagyott lesz
broker6:pwd6:a:query,config	konfigurálni tudja a figyelendő ajánlati könyveket

Az IFSC API programozása

■ **Hol találok a programozáshoz szükséges header fájlokat és függvénykönyvtárakat?**

A header fájlok (és a fejlesztést elősegítő példaprogramok forráskódja) a **k2/pkg/src** könyvtár alatt található. A header fájlok:

- *fields.h* konstans, struktúra, felsorolás és típus definíciók
- *ifsapi.h* az API függvényeinek definíciói
- *ifsdefs.h* konstans definíciók
- *ifsutil.h* mezőkezelő segédfüggvények definíciói

A dinamikus függvénykönyvtár operációs rendszertől függ. Pl.:

- k2/pkg/WIN32/ifs.dll
- k2/pkg/Linux_RHEL4/libifs.so

■ **Milyen függvények állnak rendelkezésemre az API programozásához?**

Előkészítés, be- és kijelentkezés:

- *ifsc_create, ifsc_connect, ifsc_disconnect*

Változásszámok és rekordindexek kezelése:

- *ifsc_set_get_seq, ifsc_set_get_idx, ifsc_set_get_field_seq*

Memóriatáblák lekérdezése:

- *ifsc_get_first_record, ifsc_get_next_record*
- *ifsc_get_first_idx, ifsc_get_next_idx*
- *ifsc_get_first_field_chg, ifsc_get_next_field_chg*

Ajánlati könyvek kezelése:

- *ifsc_get_orderbook_list, ifsc_orderbook_conf*
- *ifsc_get_first_orderbook, ifsc_get_next_orderbook*
- *ifsc_get_marketbyprx_list, ifsc_marketbyprx_conf*
- *ifsc_get_first_marketbyprx, ifsc_get_next_marketbyprx*

Rendszeridő és állapotmonitorozás:

- *ifsc_get_systime*

Tranzakció bevitel:

- *ifsc_orderentry, ifsc_orderentry_status_chg*
- *ifsc_mmorder, ifsc_mmorder_status_chg*

Hibakezelés és kommunikáció naplózás::

- *ifsc_get_last_errmsg*
- *ifsc_set_trace*

Továbbá még használhatjuk az *ifs_get_* és *ifs_set_* kezdetű segédfüggvényeket az üzenetek mezőinek kezeléséhez.

■ **A memóriatáblákat lekérdezhetem változásszám és fizikai index alapján. Melyiket használjam?**

A változásszám alapján történő lekérdezésnél a rekordokat frissülésük sorrendjében kapjuk meg. Ha egy rekord két lekérdezés között többször is módosult, akkor csak a legutolsó állapotot kapjuk meg. A fizikai index alapján történő lekérdezésnek gyakorlati jelentősége nincs, nem is szokták használni.

■ **Hogyan tudom monitorozni a pgw1 működését kliens oldalról?**

Az *ifsc_get_systime* függvény hívásával kapunk egy állapotrekordot, amelynek van két mezője a **K2QueryTimeOffset** és a **PgwState** a pgw1 állapotának figyeléséhez.

A **K2QueryTimeOffset** mező értéke másodpercenként nő eggyel, de lenullázódik, ha a pgw1 sikeresen befejezett egy üzenetcserét a kereskedési rendszerrel. Ha ennek a mezőnek a tartalma elkezd nagyon nőni (pl. nagyobb lesz 5-nél), akkor pgw1-nek éppen nincs kapcsolata a kereskedési rendszerrel, de még van remény, hogy automatikus újrapcsolódás után a helyzet megoldódik. Ha a mező értéke nagyobb mint 60 (eltelt 1 perc is), akkor érdemes megvizsgálni szerver oldalon a K2 processzek állapotát.

Ha a **PgwState** mező tartalma 0, akkor általában rendben fut a pgw1 processz. Ha nem 0, akkor a pgw1 leállt valamilyen hiba miatt (keresendő a *k2/log/pgw1.log* fájlban). Ez a helyzet naponta egyszer mindig előáll, amikor nap végén a BÉT üzemeltetése kijelentkezteti a felhasználókat a kereskedési rendszerből. Ha zárás előtt tapasztalható a jelenség, akkor a K2 újraindítása elkerülhetetlen.

■ **Hogyan tudom monitorozni a K2 szerver és a K2 kliens közötti kommunikációt?**

Az `ifsc_set_trace` függvénnyel lehet bekapcsolni, hogy az API egy `ifstrace.log` nevű fájlba kimásolja a szerverrel folytatott üzenetcsereket. A fájl ott keletkezik, ahol az API-t használó kliens alkalmazást indítottuk.

Ajánlat beküldés

■ **Hol találok az ordermethodset mező beállításához szükséges információkat?**

Az említett mező egy 32 hosszú karakterlánc, amely „0” és „1” karakterekből áll. A bal oldali első karaktert tekintjük a 0. (indexű) karakternek. A mező leírása a [K2_refdata1a_1_3H.pdf](#) dokumentumban olvasható.

■ **Beküldtem az ajánlatot a K2-be, de nem került be a kereskedési rendszerbe. Mi a probléma?**

Valószínűleg a K2 kliensnek nincs meg a **bypass** jogosultsága. Így az ajánlat bekerül az **orderentry** táblába, de csak „A” (accepted) státusszal. Ahhoz, hogy a K2 továbbküldje a kereskedési rendszerbe még jóvá is kell hagyni a kliens oldalról (`ifsc_orderentry_status_chg` függvénnyel lehetséges), hogy megkapja a „C” (confirmed) státuszt.

Roszsabb esetben lehetett akár bypass jogosultsága is a kliensnek, de a K2 ellenőrzés hibát talált a tranzakcióban és „D” (denied) státuszt adott.

Az említett `ifsc_orderentry_status_chg` függvénnyel nem csak jóváhagyni, hanem elutasítani is lehet egy tranzakciót, ilyenkor is „D” (denied) státuszt kap.

■ **Honnan tudhatom, hogy az ajánlatomat elfogadta a kereskedési rendszer?**

A pgw1 a „C” (confirmed) státuszú tranzakciókat egyenként beküldi (egy ciklusban legfeljebb annyi tranzakciót küld, amennyi az `orderentrypercycle` konfigurációs paraméterben rögzítve van).

A beküldés előtt „U” (unknown) státuszt ad a tranzakciónak és a válasz megérkezése után vagy „E” (entered) státuszt, ha a kereskedési rendszer befogadta, vagy „R” (refused) státuszt, ha visszautasította. „E” státusz esetén kitöltődik a **OrdNo** mező (csak STANDARD és PLUS licenc esetén) a kereskedési rendszer által kiadott ajánlat-azonosítóval. „R” státusz esetén az **Msg** mezőbe kerül a hibaüzenet.

■ **Mikor marad U státusza egy tranzakciónak és mit lehet vele kezdeni?**

Egy tranzakciónak akkor maradhat „U” (unknown) státusza, ha a pgw1 processz a beküldés után - ,de még a válasz előtt - leállt. Ilyenkor az „U” státusz a K2 újraindítása után is megmarad, de kliens oldalról lehetőségünk van a `ifsc_orderentry_status_chg` függvény segítségével ezen módosítani. Adhatunk ismét „C” (confirmed) státuszt és akkor a pgw1 újra megpróbálja beküldeni, vagy adhatunk „D” (denied) státuszt, amivel elvetjük a tranzakciót. Természetesen hagyhatjuk érintetlenül is az „U” állapotot, ami lényegében megfelel a „D” állapotnak.

■ **Az elszámolás dátumát mire állíthatom be?**

Ajánlat beküldésnél az elszámolás dátuma mezőt felesleges kitölteni, mert nem veszi figyelembe a kereskedési rendszer. Ennek a mezőnek csak a fix kötéseknel van szerepe, viszont ez a funkció a K2-ből nem érhető el. Az elszámolás dátuma mezőt a beállításoknak megfelelően a kereskedési rendszer kitölti, és ez meg is jelenik az `order` táblában.

Ajánlat módosítás (törlés+új)

■ **Mi történik egy ajánlattal, ha azt módosítom?**

Az eredeti ajánlat „A” amended státuszt kap, mely szerint érvényét veszítette, és keletkezik egy új ajánlat új azonosítóval.

■ **Tudok-e egyszerre több ajánlatot módosítani?**

Igen. Az *ifsc_orderentry* függvénynek megadható IFS_ACTION_TICKUPTICKDOWN tranzakció típus. Ilyenkor a megadott szűrési feltételeknek megfelelő ajánlatok ára módosul a megadott számú árlépésközzel.

Ajánlat visszavonás

■ **Tudok-e egyszerre több ajánlatot visszavonni?**

Igen. Szűrési feltételek megadása nélkül a cég összes ajánlata is visszavonódik! Erre nagyon kell vigyázni. Ha szűrési feltételként az ajánlat azonosító számát adjuk meg, akkor csak az törlődik, és ebben az esetben egyéb információt nem is kell megadni a visszavonáshoz.

Árjegyzői ajánlatpár

■ **Új árjegyzői ajánlatnál visszavonódik-e a korábbi?**

Igen. A kereskedési rendszerben beállított paraméterek miatt, minden esetben az új árjegyzői ajánlat visszavonja a megfelelő még nyitott árjegyzői ajánlatot.

■ **Miért utasította vissza a rendszer a beküldött árjegyzői ajánlatot?**

Csak az küldhet be árjegyzői ajánlatot, akinek ehhez jogosultsága van a rendszerben. Az éles rendszeren ezt a jogosultságot vagy a BÉT vagy az ÁKK állítja be. A tesztrendszeren csak a BÉT állíthatja be a jogosultságot, és leghamarabb a kérést követő kereskedési napon lép életbe. Egy felhasználó több termékre, illetve egy termékre több felhasználó is jegyezhet árat.

Ajánlatok lekérdezése

■ **Honnan tudom, hogy a befogadott ajánlatommal mi történt a kereskedési rendszerben?**

A beküldött ajánlat meg fog jelenni az *order* táblában, és a **Status** és **Balance** mezők figyelésével nyomon követhető az állapota. A **Status** mezőben megjelenő értékekről a **K2_refdata1a_1_3H.pdf** dokumentációban olvashat. A **Balance** mező mutatja, hogy az ajánlatból még mennyi a lekérhető mennyiség. Ha a **Status** mező „M” értéket kapott, vagyis az ajánlat teljes mértékben megkötött, akkor a **Balance** mező 0 lesz.

Kötések lekérdezése

■ **Miért kaptam meg kétszer is ugyanazt a kötés rekordot kliens oldalon?**

A K2 amikor letölti az adatokat a kereskedési rendszerből, akkor a cég saját kötéseit kétszer is megkapja. Egyszer a szűkebb adattartalmú összes kötések között, és egyszer részletesen, mint saját kötéset, vagyis amiben a cég valamelyik oldalon (vételi/eladási) szerepelt. Az első rekord újként kerül a kötések táblába, a második csak frissíti az adott rekordot, amely frissíti és növeli a változásszámot.

A K2 kliensek a változásszám alapján kérdeznak le, és a legtöbbször már a frissített rekordot kapják meg egyszer. Bizonyos körülmények között (mint például a záró-szakasz vége, amikor sok kötés keletkezik egy időben) azonban a kliens lekérdezhet úgy, hogy frissítés előtt és után is. Ilyenkor megkapja mindkétszer a rekordot.

Ajánlati könyvek figyelése

■ **Hány termék ajánlati könyvét tudom figyelni?**

A konfigurációs állományban megadott **numoforderbook** paraméter határolja be, hogy hány terméket vehetek fel figyelési listára, de mivel ez a paraméter szabadon állítható, így akár az összes terméket is figyelhetem. Két dolgot azonban fontos tudni:

- Ha túl sok termék van figyelési listán, akkor azzal jelentősen megterhelem a K2 és a kereskedési rendszer közötti kommunikációs vonalat.
- Ha PLUS licenccel rendelkezem, akkor 100 ajánlati könyv frissítése történhet egy ciklusidőn belül (ciklusidő = 1 sec). Ha ezek mellett 200 termék ajánlati könyvét akarom figyelni, akkor $200/100 = 2$ másodpercenként fognak az ajánlati könyvek frissülni.

■ **Milyen mélységben láthatom a különböző ajánlati könyveket?**

Az 1-es kereskedési rendszerünk mindkét ajánlati könyv típus esetében legfeljebb 20-as mélységet tud szolgáltatni, természetesen külön vételi és eladási oldalon. A 2-es rendszerben nincs ilyen korlát.

Egyéb kérdések

■ **Miben tud többet vagy kevesebbet a K2 a Kereskedési munkaállomás (TW) szoftvernél?**

A K2 segítségével a brókercégek saját BackOffice rendszere kapcsolható össze a BÉT kereskedési rendszerével, lehetőséget adva egy közvetlenebb kereskedési technikára. Így lehetőség van például automatikus árjegyzésre is. Tehát olyan rendszer építhető a segítségével, amellyel sokkal gyorsabban lehet reagálni a piaci változásokra.

A kereskedési munkaállomás szoftver (TW) leginkább a kézi vezérlésű kereskedést támogatja a maga grafikus felületén keresztül. De rendelkezik olyan plusz funkciókkal is, amelyek a K2-ből nem érhetők el, mint például a kereskedési számlák kezelése, vagy fix ajánlatok beküldése. A kereskedési számlák kezelése egy olyan funkció, ami miatt a csak K2-ön keresztül kereskedő szekciótárgok is kénytelenek TW szoftvert telepíteni és használni. Tehát a K2 első használata előtt a TW segítségével hozzá kell rendelni a szükséges kereskedési számlákat a K2-es felhasználóhoz, mely előfeltétele a sikeres tranzakció bevitelnek.