
K2 PROGRAMOZÁSI SEGÉD-DOKUMENTÁCIÓ

Cím: Programozási segéd dokumentáció
Készítette: Informatikai Fejlesztés
Ellenőrizte:
Verzió: 1.1
Dátum: 2002. május 16.

Fájlnév:Programozasi segédlet.doc

Tartalom

1. Bevezető	3
2. Kapcsolódó dokumentációk	3
3. Hasznos információk	3
3.1. Mit tud csinálni egy K2 kliens? Mi a kliens szerepe?	4
4. Programozási / K2 kliens fejlesztési tanácsok	4
4.1. Támogatott szerver és kliens platformok	4
4.2. Ajánlatbevitel	5
4.2.1. Ajánlatok státuszai a K2-ben	5
4.2.2. Ajánlatok státuszai az MMTS rendszerben	6
4.2.3. A bevitt ajánlatokat a K2 is sorszámozza	6
4.2.4. Bevitt ajánlatok saját megjelölése/azonosítása	6
4.2.5. Ajánlatok bevitelének strukturális és tartalmi ellenőrzései	6
4.2.6. Az ajánlatbevitelhez szükséges kereskedési számlaszámok	6
4.2.7. Az ár és hozam mezők kezelése ajánlatbevitelén	6
4.2.8. Az értékpapír tábla (BoardID), értékpapír (SecID) és SecBoardID azonosítók kezelése	6
4.3. Információ lekérdezés a K2-ből, információ típusok	6
4.3.1. K2 kliens adatlekérdezések típusai	6
4.3.2. Gyorsabb kliens lekérdezés	6
4.3.3. Adatlekérdezési függvények visszatérési értékei	6
4.3.4. Változásszámok táblánként	6
4.4. Kliens ellenőrzés: milyen K2 szerverre kapcsolódik	6
4.5. Egy K2 válasz feldolgozása	6
4.6. Minden kliens függvényhívás után hibaérték és hibaüzenet vizsgálat	6
4.7. K2 szerver: napi indítás, kapcsolat megszakadása, normális napvégi leállítás, hiba esetén történő leállítás	6
4.7.1. A K2 napi normális indítása	6
4.7.2. A K2 napközbeni újraindítása esetén	6
4.7.3. A K2 napvégi rendes leállása	6
4.8. Több K2 kliens processz és szálprogramozás	6
4.9. Egy javasolt napi ciklus a K2-vel való kommunikációra	6
4.10. Nyomkövetés a kliens oldalon	6
4.11. Példaprogramok, hasznos utility-k	6
4.12. Új papírokra való figyelés és más kereskedési paraméterek változása	6
5. A K2 napi üzemeltetésével kapcsolatos egyéb tudnivalók	6
5.1. A K2 reggeli indítása és esti megállítása	6
5.2. K2 kliens felhasználói adatok kézi beállítása	6
5.3. A K2 idejének időszakos beállítása	6

Módosítások jegyzéke

Verzió	Szerző	Dátum	Módosítások összefoglalása
1.0	Pittner Ferenc	2002. május 9	Első verzió
1.1	Vass Attila	2002. május 16.	Kiegészítések

1. Bevezető

A Budapesti Értéktőzsde (BÉT) 2002. májusában megvásárolta az Effice Mérnöki iroda Kft-től a K2 kapcsolati kiszolgáló szoftver forráskódját, annak módosítási és értékesítési jogát. A tranzakció eredményeként lehetővé vált az, hogy a K2 szoftvert a BÉT értékesítse és tartsa karban. Jelen dokumentummal programozási szempontból kívánunk segítséget nyújtani mindazon brókercégek részére, akik a K2 szoftverhez kliens kapcsolódási felületet kívánnak fejleszteni.

Ez a dokumentum azért született, mert az szerzők által készített eredeti interfész leírás és K2 Felhasználó kézikönyv több olyan részletre nem hívja fel a fejlesztők figyelmét, amely véleményünk szerint elengedhetetlen a kliens szoftver pontos fejlesztési tervezéséhez.

Megtörténhet, hogy a későbbiekben a BÉT munkatársai módosítják az Effice által készített dokumentációkat, és az itt leírt információk átkerülnek a Felhasználói kézikönyvbe.

Javasoljuk a K2 kliens fejlesztőknek, hogy fejlesztés előtt alaposan olvassák el mind jelen dokumentumot, mind pedig a K2 Felhasználói kézikönyvét. Fontos, hogy a K2 kliens szoftver tervezése előtt megértsék a K2 működési logikáját, mert így érhető csak el egy minden körülmények között biztonságosan üzemelő szoftvert fejlesztése.

Ebben a dokumentumban hasznos információkat foglaltunk össze, és megfogalmaztunk a K2 kliens működésével kapcsolatos ajánlásainkat, amelyek természetesen nem kötelező jellegűek. Ezt a dokumentumot a BÉT munkatársai a K2 auditálása, és átvétele alatt szerzett tapasztalatok alapján készítették.

2. Kapcsolódó dokumentációk

A dokumentum elkészítésének időpontjában az alábbi verziójú dokumentációkra hivatkozunk:

- K2/ifsc Felhasználói kézikönyv (v2.2/2001) - Effice
- MMTSI adattípusok (v1.1/2001) – Effice
- MMTSII adattípusok (v1.1/2001) – Effice

3. Hasznos információk

A dokumentum készítésekor a K2 a létező egyetlen fizikai implementációja a Távkereskedési Szabályzatban definiált Kapcsolati Kiszolgálónak. Megvalósítja az

automatizált/programozható kommunikációt a brókeri informatikai rendszer és a BÉT MMTS kereskedési rendszere között.

3.1. Mit tud csinálni egy K2 kliens? Mi a kliens szerepe?

A **K2 szerver** egy külön gépen fut és kapcsolatot teremt egyrészt az MMTS-el, másrészt pedig a bróker cég által fejlesztett / megvásárolt K2 kliens alkalmazással. A K2 lekérdezi a kereskedési adatok nagy részét a központi rendszertől, valamint továbbítja a központi kereskedési rendszernek a K2 kliens felől érkező ajánlatokat. Maga ez a szerver komponens vásárolható meg mint K2. Bár a K2 szerverhez több K2 kliens is csatlakozhat a brókeri rendszerből (lásd lentebb), maga a K2 szerver egyetlen felhasználóként jelentkezik be a BÉT központi rendszerébe, és a K2 kliensek által bevitt ajánlatok mind ennek az egy felhasználónak a nevében fognak beérkezni a központba.

A **K2 kliens** nem más mint egy alkalmazás / szoftver, amelyet a bróker saját maga fejleszt, vagy fejlesztet. A kliens szoftver a K2 szerverhez kapcsolódik és azzal tartja a kapcsolatot. A K2 kliens lehet a bróker saját rendszerének integráns része is, amelyet felkészítettek a K2-vel való kapcsolattartásra. A 2. pontban felsorolt dokumentációk a K2 klienst IFSC-nek hívják (Interface Socket Client), és több helyen mint felhasználó (F) jelenik meg. A K2 kliens a különböző platformokra lefordított kliens rutinkönyvtárak (DLL vagy LIB) segítségével tudják hálózaton keresztül tartani a kapcsolatot a K2 szerverrel.

A BÉT központi kereskedési rendszerének sajátosságai miatt fizikailag a K2 szerverből két különálló létezik (egy azonnali és egy derivatív modul), ami két párhuzamosan futó programot is jelent. Kliens oldalon azonban csak egyetlen rutinkönyvtár van, és a kliens szoftver a felkapcsolódáskor mondja meg, hogy ő most az azonnali, vagy a derivatív K2 szerver programhoz szeretne-e kapcsolódni, sőt mindkét rendszerhez egyszerre is kapcsolódhat

A K2 kliens tehát alapvetően ajánlatokat tud beküldeni a K2 szerveren keresztül a központi kereskedési rendszerbe, valamint a piaci adatokat tudja lekérdezni a K2 szervertől. A kapcsolat jellegzetesen kérdezz-felelek, ugyanis a K2 szerver csak a kliens kérdéseire válaszol, magától nem küld semmilyen adatot. Egy K2-re több kliens alkalmazás is csatlakozhat, de ezzel kapcsolatban vannak megkötések (lásd 4.8, Több K2 kliens processz és szálprogramozás).

4. Programozási / K2 kliens fejlesztési tanácsok

4.1. Támogatott szerver és kliens platformok

Jelenleg a K2 szerver HP UNIX, és LINUX alatt érhető el, a linuxos környezet természetesen olcsóbb, ezért alapvetően ennek a verzióknak az értékesítésére készült fel a BÉT. A megcélzott Linux típusa a RedHat.

A kliens oldalon több megoldás közül választhat a fejlesztő:

- Windows NT;
- Linux (RedHat)
- HP-UX

Kliens oldalon egy függvénykönyvtár (és az ehhez szükséges állományok) állnak a fejlesztők rendelkezésére. Ez Windows alatt egy DLL és pár header fájl, UNIX/Linux alatt LIB könyvtár és header fájlok.

Elképzeltető más platformokra is lefordítani a kliens függvénykönyvtárat, de erre konkrét igény alapján kerülhet csak sor, és nincs garantálva az eredmény.

4.2. Ajánlatbevitel

Fontos részletesen megvizsgálni és megérteni az ajánlatbevitel folyamatát. Annak ellenére, hogy ezt a témát a K2/ifsc Felhasználói kézikönyv némileg tárgyalja, fontosnak tartjuk az alábbiakat:

4.2.1. Ajánlatok státuszai a K2-ben

- a. Ajánlatokat bevitelét, módosítását törlését természetesen a K2 kliens kezdeményezi.
- b. Ezek az igények a kliensről bekerülnek először a K2-be, és a kliens erről visszaigazolást+azonosítót is kap. (Az ajánlat lehetséges státuszai a K2-ben: *accepted* – még jóvá kell hagyni; *confirmed* - jóváhagyás nem szükséges; *denied*-hibás ajánlat tartalom).
- c. Ha az ajánlatot még egy másik kliensnek jóvá kell hagynia, akkor az addig várakozik a K2-ben, míg ez meg nem történik (addig nem küldi el a K2 a központi kereskedési rendszernek). (Az ajánlat lehetséges státuszai a K2-ben a művelet után: *confirmed* - jóváhagyás megvolt; *denied*-nem lett jóváhagyva).
- d. A K2 szerver ciklusban ellenőrzi, hogy van-e ezzel kapcsolatban tennivaló, és ha talál várakozó ajánlatot, akkor azt elküldi a központi kereskedési rendszernek, és erre valószínűleg (figyelem!) választ is kap. (Az ajánlat lehetséges státuszai a K2-ben: *unknown* – el lett küldve a kereskedési rendszernek de még nem jött válasz; *refused* – a központi rendszer visszautasította valamiért; *entered*-sikeres ajánlatbevitel a központi kereskedési rendszerbe).
- e. A kliens erről (d. pont) nem kap automatikusan visszajelzést, hanem saját magának kell lekérdeznie az ajánlatok tábláját, és ha talál új adatot, akkor azt fel kell dolgozni, illetve folyamatosan figyelni kell az ajánlatok státuszváltozásait (ez a feladat nem túl nehéz, mivel a státuszváltozások nyilvántartása is a változásszámok kezelésével történik lásd 4.3.4).

A valós életben előállhat még egy eset a d. és e. pont között: ha a K2 valamiért leáll rögtön a központi rendszernek történő ajánlatbeküldés után, tehát pár ajánlat U státuszban marad (ez azt jelenti, hogy a K2 leakadás előtt megpróbálta elküldeni a központi rendszernek, de ez vagy nem sikerült, vagy pedig a K2 már nem kapta meg a választ). Ilyenkor az ilyen ajánlatokkal újraindításnál nem csinál semmit a K2, ezért a kliens oldalon kell eldönteni,

hogy áteszi őket Confirmed-be (ilyenkor ismét elküldi őket a központi kereskedési rendszernek a K2), vagy Denied-ba (ilyenkor nem küldi el őket a K2 a központi rendszernek) – lásd 4.7.2 A K2 napközbeni újraindítása esetén.

Már itt megjegyezzük, hogy az ajánlatok státuszainak változását a K2 egy-egy fájlban tartja nyilván, amelyeket az rldump1 és rldump2 segéd programokkal lehet megtekinteni (4.11, Példaprogramok). Ez tesztelésnél igencsak hasznos.

Ugyancsak fontos, hogy az ajánlatok státuszát a K2 is állítja és maga a kliens alkalmazás is állíthatja (c.).

A Felhasználói kézikönyvben elég zavaró elnevezések láttak napvilágot. Például nagyon fontos tudni azt, hogy ajánlatokat a kliens a K2-től két helyről/táblából kérdezhet le:

- a K2 saját ajánlati táblája (pl. 11. Tábla Ajánlatbeviteli rekord, IFSC Felhasználói kézikönyv);
- az MMTS-ből kapott ajánlatok (a K2 másik ajánlati táblája, pl. 8. Tábla Ajánlat rekord, IFSC Felhasználói kézikönyv).

Az első táblában megtalálhatóak a fentebb taglalt státuszok, és a b. pontban említett K2 azonosító (pl. 11. táblázat STATUS és ORDERID mezői).

4.2.2. Ajánlatok státuszai az MMTS rendszerben

Az ajánlatoknak természetesen a központi kereskedési rendszer is ad egy állapotot (pl. OPEN, AMENDED, MATCHED, stb. lásd MMTS I. és MMTS II. adattípusok - Kapcsolódó dokumentációk, 2.). A központi kereskedési rendszer által adott státuszok például a 8. táblázat STATUS mezőjében található.

4.2.3. A bevitt ajánlatokat a K2 is sorszámozza

Az a. pontban bevitt ajánlatnak – amennyiben ez sikeresen bekerül a K2-be – a K2 ad egy saját sorszámot (ifsc_orderentry függvény new_orderid mezője). Ez két okból fontos:

- ha megváltozik az ajánlat státusza (pl. d. pontban leírtak miatt), akkor a kliens egyszerűen be tudja azonosítani, hogy melyik ajánlatnak változott meg az állapota;
- ismerve a már elfogadott ajánlatok K2 sorszámát, egy esetleges K2 újraindításkor könnyű aktualizálni az új és/vagy módosult ajánlatok státuszát a kliens oldalon is (hogy szinkronban legyen az újraindult K2-vel, lásd 4.7).

Az ajánlatokat egyébként a központi kereskedési rendszer is egyedileg megjelöli. A megjelölés különbözik az azonnali és a derivatív rendszerben (8. tábla OrdNo mezője egész típusú, míg a derivatív 27. tábla OrdNo mezője string típusú). Ezek az azonosítók is megtalálhatók az ajánlatból esetlegesen született üzlet vételi vagy eladási oldalán.

4.2.4. Bevitt ajánlatok saját megjelölése/azonosítása

Javasoljuk az ajánlatok esetében a BROKERREF mezők használatát, mert ez a mező végigkíséri az ajánlatot egészen az ajánlatból született üzletig (megjelenik annak vételi vagy eladási oldalán). Ezzel lehet a háttérrendszerből érkező egyedi azonosítót adni egy-egy ajánlatnak és utólag vizsgálni az adott ajánlat sorsát. Ez teremti meg tehát a kapcsolatot a kliens rendszer által kezdeményezett ajánlat és a központi rendszerből érkező üzlet között.

4.2.5. Ajánlatok bevitelének strukturális és tartalmi ellenőrzései

A kliens által bevitt ajánlatok több helyen is (3!) ellenőrzésre kerülnek és elakadhatnak:

- a K2 kliens függvények egy gyors ellenőrzést hajtanak végre (még a K2 szervernek való elküldés előtt): ha az ajánlat nem megfelelő, akkor az *ifsc_orderentry* függvény negatív értékkel tér vissza és a hiba egy másik függvénnyel lekérdezhető.
- az ajánlatot ellenőrzi tartalmi szempontból a K2 szerver: hiba esetén feltölti pl. a 11-es táblázat (IFSC Felhasználói kézikönyv) Msg mezőjét egy hibaüzenettel, és az ajánlat K2 státuszát DENIED-re állítja. Erről a kliens csak akkor szerez tudomást, ha ezt a táblát lekérdezi és az állapotot vizsgálja.
- a központi kereskedési rendszer nem fogadja el az ajánlatot (pl. azért mert a kereskedést felfüggesztették), akkor ez bekerül az előbb említett 11-es és 8-as táblázatokba is.

4.2.6. Az ajánlatbevitelhez szükséges kereskedési számlaszámok

Az ajánlatbevitelhez szükségesek a kereskedési számlaszámok (trading account), ezeket nem lehet lekérdezni K2-n keresztül, tehát javasoljuk, hogy a kliens háttérrendszerében ezek kerüljenek eltárolásra. A kereskedési számlaszámok használata eltér az azonnali és a derivatív kereskedési rendszerben, sőt, a derivatív rendszerben ezek naponta változnak, ugyanis ezeket a BÉT naponta automatikusan generálja a Kelertől kapott pozíciók alapján. Ezért erre fokozottan ügyelni kell, ugyanis rossz kereskedési számlaszám esetén a központi rendszer az ajánlatot elutasítja.

Az azonnali rendszer számlaszámának felépítése:

példa: 0011-0000001

- ahol az első számjegy a BÉT-es, Keler-es tagkódot jelenti, a második szám egy sorszám.
- az azonnali rendszerben a számlaszámok nem változnak (a bróker cég kérheti ezek bővítését vagy törlését).

A derivatív kereskedési rendszer számlaszámainak felépítése:

Például 7-006-I. A kereskedési számlák felépítése három részből áll „-”-al elválasztva:

I. rész:

ügyfél keler száma (max. 6 pozíció, elővezető nullák nélkül). Például 1,13,235. Vagy

cég keler száma (2 pozíció, ahol az első mindig nulla). Például 01,02,03. Vagy

speciális ügyfélszámla (6 db "8"-as karakter, utána a cég kódja, majd "I") - a még kóddal nem rendelkező új ügyfelek részére

II. rész:

tőzsdetag Keler tagkódja (3 pozíció, elővezető nullákkal)

III. rész:

I vagy **H** vagy **M**, (I-ügyfélszámla, H-saját szám-la, M-market maker)

- a derivatív rendszerben a számlaszámok naponta változnak a Kelertől este a BÉT részére átküldött adatok alapján.

4.2.7. Az ár és hozam mezők kezelése ajánlatbevitelén

A dokumentáció nem szól részletesen arról, hogy az ajánlatbevitelnél hogyan kell kezelni az ár és mennyiség mezőket, pedig ezek nemcsak speciálisak, hanem különböznek is az azonnali és a derivatív rendszerben.

Az azonnali rendszerben mind az ár, a hozam és a mennyiség is egész számokként kerülnek kitöltésre: a tizedes jegyeket tartalmazó árakat egyszerűen fel kell szorozni annyi tizedessel, ahánnyal az illető papírt kereskedik. A szorzótényező az illető papír egyik jellemzője (5. számú Értékpapír tábla, PriceDecimals, YieldDecimals). Vigyázat ezeket a BÉT egyoldalúan átállíthatja egyik napról a másikra! Tehát ezeket célszerű minden K2 indításkor felolvasni és eltárolni. Főleg olyan rendszereknél fontos erre vigyázni, amelyek „állandóan mennek” és nem indítás/leállítás üzemmódban működnek.

A derivatív rendszerben az ár és hozam mezők speciális szerkezetűek IFS_FIXREAL amelyik két részből áll:

- az első mező egy valós szám (double) magának az árnak vagy hozamnak az valós értékét tartalmazza kerekítés nélkül
- a második mező pedig megadja a tizedesek számát.

4.2.8. Az értékpapír tábla (BoardID), értékpapír (SecID) és SecBoardID azonosítók kezelése

Felhívjuk a figyelmet arra, hogy az értékpapírtábla azonosítóját (BoardID), és a papír azonosítóját (SecID vagy SecCode) a K2 kliens oldalán többféleképpen is kezelik (az MMTS sajátosságai miatt):

- ajánlatbevitelnél két külön mezőben kell megadni az értékpapírtáblát (pl. FŐRÉ) és az értékpapír azonosítóját (pl. MOL) – lásd pl. 11.tábla a Felhasználói kézikönyvben;
- a sikeresen bevitt ajánlatok lekérdezésénél az értékpapírtábla és a papír neve egyetlen mezőben érkezik vissza SecBoardID, az azonnali rendszerben egy felvezető „0” (nulla ASCII 48-as karakterrel) (pl. „0FŐRÉ MOL”) lásd Felhasználói kézikönyv 8. táblázat. A derivatív rendszerben nincs felvezető „0” karakter.

További információk találhatóak az MMTS I és MMTS II adattípusok dokumentumokban.

4.3. Információ lekérdezés a K2-ből, információ típusok

4.3.1. K2 kliens adatlekérdezések típusai

A K2 kliens amikor adatokat kérdez le, akkor alapvetően négy fajta táblával találja magát szemben:

- statikus tábla: amikor a táblának a tartalmát elég egyszer lekérdezni, mert ez napközben nem módosul;
- dinamikusan hízó tábla: amikor a tábla folyamatosan bővül új rekordokkal (és a meglévők nem módosulnak), ilyen pl. az üzletkötések táblája;
- táblák amelyek tartalmazznak statikus és dinamikus adatokat is: ilyen például az értékpapírok táblája (5. és 24. táblázat), amelyeknek az eleje statikus, a többi dinamikus (a „belseje változik a táblának”).
- minden változásnál akár teljes mértékben változó táblák: ilyenek az ajánlati könyvek

Az adatok lekérdezésére is több lehetőség áll a K2 kliens rendelkezésére:

- egy tábla rekordjainak a **fizikai rekord sorszám szerinti** lekérdezése (idx): ifs_get_first_idx és ifs_get_next_idx függvények. Ezzel tehát meg lehet például tudni, hogy a táblában mi a 16. rekord (sor).
- egy tábla megváltozott tartalmú rekordjainak a lekérdezése **változásszám alapján** (record): ifs_get_first_record és ifs_get_next_record függvények. A K2 minden tábla minden rekordjára tárol egy változásszámot, amelyet növel ha a rekord módosult. Ha a kliens emlékezik arra, hogy legutóbb melyik változásszámú adatot kapta meg, akkor meg tudja kérdezni a ifs_get_next_record függvénnyel a következő megváltozott rekordot.
- az **értékpapír táblánál** mivel sok mező van, az egyes mezőknek a rekordokban külön-külön van változásszámuk, és ez alapján le lehet kérdezni a **megváltozott mezők** tartalmát (tehát ez egy fajta sáv szélesség-kihasználás optimalizálás) (field): ifsc_get_first_field_chg és ifsc_get_next_field_chg.
- az **ajánlati könyvek lekérdezése** speciális módon, külön függvényekkel történik (orderbook és marketbyprx): kliens oldalon először be kell állítani, hogy milyen ajánlati könyveket figyeljen a K2 (ifs_orderbook_conf és ifsc_marketbyprx_conf), majd az ajánlati könyveket az ifsc_get_first_orderbook, ifsc_get_next_orderbook, ifsc_get_first_marketbyprx ifsc_get_next_marketbyprx függvényekkel lehet az adatokat lekérdezni.

4.3.2. Gyorsabb kliens lekérdezés

A K2 kliens programok tudják gyorsítani az adatok lekérdezését a K2 szervertől, ha egy tábla (pl. ajánlatok) összes változását lekérdezik, amíg nincs több adat az illető táblában, és csak utána kezdik el lekérdezni a következő táblát. Tehát nem azt teszik például, hogy lekérdeznek egy rekordot az ajánlatokból, utána lekérdeznek egy rekordot a kötésekben és így tovább. Ez a működés abból adódik, hogy a K2 IFSC kliens függvénnyel mindig csak egyet kérdez, de az IFSS erre egy csomóban válaszol. Azaz megelőlegezi a következő

rekordokat is, és beteszi ezeket egy bufferbe, majd a következő IFSC kliens kérést az IFSC már ebből a saját bufferből szolgálja ki.

Ha a kliens nem a következő rekordot kérdezi le ugyanabból az MMTS-ből (1 v. 2), akkor a buffert tartalmát eldobja.

4.3.3. Adatlekérdezési függvények visszatérési értékei

A dokumentáció ugyan említi, de nagyon fontos, hogy minden esetben értelmezzük a K2 kliens oldalon az adatlekérdezési függvények visszatérési értékeit (pl. `ifsc_get_next_record`):

- ha a visszatérési érték 0 (nulla), akkor sikeres volt a függvény;
- negatív érték hibára utal, le kell kérdezni feltétlenül a hibát a `get_last_errormsg` függvénnyel, mert a hiba értéke a következő függvény hívásnál felülíródik, ha az is hibával tér vissza;
- ha a visszatérési érték `IFS_NOMORE`, akkor nincs több adat jelenleg a táblában;

4.3.4. Változásszámok táblánként

Amint már volt róla szó, minden K2-ben nyilvántartott táblában rekordok vannak és mindegyiknek van egy jellemzője: a változás száma. Ez induláskor egy sorszám, amelyekből a legnagyobbat a K2 mindig nyilvántartja és ez lesz a tábla változásszáma. Amikor egy rekord megváltozik, akkor a tábla változásszáma nő eggyel és az illető rekord változásszámát erre az értékre állítja a K2 szerver. A kliens amikor lekérdez egy rekordot, megkapja annak változásszámát, ezért legközelebb tudja, hogy azokra a rekordokra van szüksége, amelyeknek a változásszáma nagyobb annál, mint ami neki van.

Az értékpapír táblánál az egyes mezőknek külön is van változásszámuk.

4.4. *Kliens ellenőrzés: milyen K2 szerverre kapcsolódik*

A kliensnek több ellenőrzési lehetősége is van a K2 szerverhez való kapcsolódáskor (`ifsc_connect` függvény):

- **mmts_type** mező: megkapja, hogy MMTSI vagy MMTS II K2 szerverhez kapcsolódott-e. Ezt célszerű ellenőrizni, ugyanis azt, hogy milyen modulhoz csatlakozzon a kliens azt az `ifsc_create` `hostname` & `service_name` paramétereiben tudja megadni (és ezt esetleg elállíthatják).
- **pid** mező: a K2 szoftver azonosítója, ezzel lehet ellenőrizni, hogy a korábban kapott pid érték megegyezik-e a most kapottal, mert ha nem, akkor a kliens adatokat és a szerver táblákat érdemes szinkronba hozni (teljes újbóli letöltéssel). Ha a két pid érték egyezik, akkor elegendő a kliens oldalról az egyes táblák változásszámainak beállítása, és a lekérdezések onnan folytatása, ahol a kliens legutóbb abbahagyta. megváltozott pid érték azt jelenti, hogy a K2-t újraindították.
- **tradeid** mező: a központi kereskedési rendszer azonosítója. Használatára ugyanaz érvényes mint a pid használatára.

4.5. Egy K2 válasz feldolgozása

A K2 kliens karaktersorozat alapú üzenetekkel kommunikál a szerverrel. Azt, hogy a kliensnek az adatokat hogyan kell oda-vissza konvertálni, a Felhasználói kézikönyv csak a végén említi meg. Mindenképpen javasoljuk a példaprogramok áttekintését fejlesztés előtt (lásd 4.11, Példaprogramok) és a mezőkódoló, meződekódoló és mezőléptető segédfüggvények használatát (lásd Felhasználói kézikönyv).

4.6. Minden kliens függvényhívás után hibaérték és hibaüzenet vizsgálat

Nagyon fontos és kifejezetten javasoljuk – mint azt a 4.3.3, Adatlekérdezési függvények visszatérési értékei részben már taglaltuk – hogy minden ifsc függvényhívás visszatérési értékét a kliens program megfelelően kezelje le. Általános szabály, hogy ha a függvény negatív értékkel tér vissza, akkor hiba történt, és ilyenkor le kell kérdezni a hiba szövegét is.

4.7. K2 szerver: napi indítás, kapcsolat megszakadása, normális napvégi leállítás, hiba esetén történő leállítás

Külön figyelmet és tervezést jelent a kliens oldalon a K2 indítása, újraindítása, leállása és a kereskedés vége. A K2 szerver két külön részből áll:

- a PGW tartja a kapcsolatot a központi kereskedési rendszerrel
- az IFSS tartja a kapcsolatot a K2 kliens alkalmazásokkal.

A K2 szervert nap elején el kell indítani, a nap végén pedig le kell állítani, illetve a kereskedés végén vagy hosszabb vonalszakadás esetén a PGW magától leáll. Mit tud tenni és mit kell tennie ilyenkor a kliensnek?

4.7.1. A K2 napi normális indítása

A K2 reggeli indulása után a következő tennivalók vannak kliens oldalon (többek között):

- megjegyezni a kapcsolat paramétereit (lásd 4.4).
- megjegyezni a táblák változásszámait
- felprogramozni, hogy milyen ajánlati könyveket akar látni a kliens (ifsc_orderbook_conf és ifsc_marketbyprx_conf függvények).

4.7.2. A K2 napközbeni újraindítása esetén

Ha a K2-t teljesen újra kell indítani (huzamosabb vonalszakadás vagy a központi rendszer leállása miatt), akkor a kliensnek:

- újra kell kapcsolódni a K2 szerverhez
- ellenőrizni kell a kapcsolat paramétereit (lásd 4.4).
- letölteni ismét teljesen az összes adatot vagy beállítani a táblák korábbi változásszámait és onnan folytatni a lekérdezést (attól függően, hogy milyen újraindítás történt).
- felprogramozni ismét, hogy milyen ajánlati könyveket akar látni a kliens (`ifs_orderbook_conf` és `ifsc_marketbyprx_conf` függvények), és letölteni ezek adatait.
- korábban beküldött ajánlatok státuszainak szinkronizálása, U (unknown) státuszú ajánlatok kezelése.

Az utolsó pont napi eljárásrend szempontjából elég fontos: ha a K2 valamiért leáll, és pár ajánlat U státuszban maradt: ez azt jelenti, hogy a K2 leakadás előtt megpróbálta elküldeni a központi rendszernek, de ez vagy nem sikerült, vagy pedig a K2 már nem kapta meg a választ. Ilyenkor az ilyen ajánlatokkal újraindításnál nem csinál semmit a K2, ezért a kliens oldalnak kell eldönteni, hogy átteszi őket Confirmed-be (ilyenkor ismét elküldi őket a központi kereskedési rendszernek a K2), vagy Denied-ba (ilyenkor nem küldi el őket a K2 a központi rendszernek).

Látható, hogy az újraindítás miatt jelentős többletfeladatok vannak.

4.7.3. A K2 napvégi rendes leállása

A kereskedés végeztével, mivel a BÉT a központi kereskedési rendszert leállítja, a PGW processz egy idő után magától megáll a K2-n. A K2 kliens ilyenkor le tudja még kérdezni a K2-ben lévő adatokat, de természetesen ajánlatok bevitelére már nem kerülhet sor. A kliensnek lehetősége van arra, hogy lekérdezze azt, hogy a PGW rendszeren működik-e, tehát információt kap arról, hogy van-e kapcsolat a központi kereskedési rendszerrel.

Ezt az információt a kliens az `ifsc_get_systime` függvénnyel szerezheti be. A `K2QueryTimeOffset` mező megadja, hogy hány másodperce volt utoljára sikeres kapcsolatfelvétel a központi rendszerrel. Itt javasoljuk a következő értékeket figyelni, illetve státuszokat közölni a brókerrel:

- ha `K2QueryTimeOffset` mező értéke < 10 mp, akkor normális napi működésnek tekinthető
- ha $10 \leq \text{K2QueryTimeOffset} < 180$ mp, akkor átmenetileg nincs kapcsolat a központi kereskedési rendszerrel;
- ha `K2QueryTimeOffset` > 180 mp, akkor a központi kereskedési rendszerrel való kapcsolat nagy valószínűséggel végleg megszakadt.

Ugyancsak hasznos információt hordoz a `PGWSTATE` mező is, ugyanis ha ez `false`, akkor azt jelenti, hogy a PGW már nem fut, és kézi beavatkozás szükséges a K2 újraindításához vagy leállításához.

4.8. Több K2 kliens processz és szálprogramozás

A K2 szoftver megismerésénél kiderült, hogy programozási szempontból kifejezetten tiltott, hogy a kliens oldalon egyetlen kliens program több többszörös programozással több kapcsolatot hozzon létre ugyanazzal a K2 szerverrel. Arról van szó, hogy az IFSC által a kliens oldalon használt adatpufferből egy kliens alkalmazáshoz egy tartozik. Ennek következtében kifejezetten káros lenne az, ha az egyik szál az adatlekérdezést nem tudja teljesen befejezni, és egy másik szál másik adatlekérdezést indít, ugyanis ilyen esetben felülíródik az első szál által használt adatterület. Az egyedi adatlekérdezéseket tehát be kell teljesen fejezni, mielőtt egy másikat kezdeményez a kliens.

Annak nincs akadálya tehát, hogy egy alkalmazás több kapcsolatot hozzon létre a K2 szerverrel, csak ezt ne szálprogramozással csinálja, hanem egy másik ciklusban. Annak sincs akadálya, hogy több párhuzamosan futó alkalmazás külön-külön hozzon létre kapcsolatokat a K2 szerverrel.

4.9. Egy javasolt napi ciklus a K2-vel való kommunikációra

A fentiekből kitűnik, hogy nem teljesen triviális a kliens oldalról a K2 szerverrel tartani a kapcsolatot, főleg abban az esetben ha valami hiba van, esetleg a szerver újraindul. A javasolt folyamatábrát a dokumentum végén találhatják.

4.10. Nyomkövetés a kliens oldalon

A kliens oldali fejlesztést megkönnyítheti ha felmerülnek kétségek, hogy pontosan milyen adatot küld a kliens a K2 szervernek, ha a kliens oldalon átmenetileg bekapcsolják a nyomkövetést az `ifs_set_trace` függvényvel. Ennek hatására a K2 kliens a szervernek küldött üzeneteket és a kapott válaszokat szöveges formában eltárolja egy `ifstrace.log` állományba. Az állomány szerkezetéről a fejlesztők tudnak felvilágosítást adni.

4.11. Példaprogramok, hasznos utility-k

A fejlesztés során megkönnyítheti a munkát, ha megtekintjük a példaprogramokat, amelyeket a rendszerhez mellékeltek, ugyanis ezek C forráskódban is megtalálhatók:

- `demo.c`: ifsc (k2 kliens) példaprogram
- `gen_order`, `get_ob`, `get_table_ob_watch`, `order_status`: példaprogramok az ajánlat bevitelre, az ajánlati könyv lekérdezésére, a tábla lekérdezésre, és az ajánlat jóváhagyására

Kiemelt fontosságúak az **rldump1** és **rldump2** programok. Ezek parancssorból indíthatók, és segítségével lehet megtekinteni a szerveren azt, hogy a kliens oldalról beküldött ajánlatok milyen státuszváltozásokon mentek keresztül. Fejlesztés folyamán hasznos lehet. A programok természetesen a szerveren található a bin könyvtárban, a keresendő log állományok pedig a log könyvtárban

4.12. Új papírokra való figyelés és más kereskedési paraméterek változása

A K2 kliens fejlesztése folyamán fokozottan figyelni kell arra, hogy a központi rendszer szinte teljes mértékben paraméterezett, és ezek a paraméterek megváltozhatnak. Megváltozhat például a papírok neve, a kereskedett tizedesjegyek száma az árban, új papírok jöhetnek be, kitolódhat a kereskedési idő, stb. Ilyenkor arra kell törekedni, hogy a kliens, amennyire csak lehetséges, használja a K2-ből letölthető adatokat - és ne a saját rendszerében találhatóakat használja, valamint ezeket minden indításkor / csatlakozáskor frissítse.

5. A K2 napi üzemeltetésével kapcsolatos egyéb tudnivalók

5.1. A K2 reggeli indítása és esti megállítása

Említettük a fentiekben, hogy a K2 logikája szerint a K2 szervert üzemeltetés szempontjából reggel el kell indítani és este le kell állítani. A kereskedés végeztével a PGW processz magától leáll. Ilyenkor a kliens programok még kapcsolatban maradnak a K2-vel (az nem szakad meg), adatokat lehet lekérdezni. Az IFSC processzek leállítását a szerveren manuálisan kell elvégezni a k2stop paranccsal. Ilyenkor természetesen a kliens kommunikáció is megszakad a K2 szerverrel.

5.2. K2 kliens felhasználói adatok kézi beállítása

Azt, hogy milyen néven és jelszóval lehet bejelentkezni a K2-be kliens oldalról, a szerver egy isfc.uaf fájlban tárolja. Ennek a fájlnak a bővítése és módosítása kézzel történik, jelenleg nincs rá automatizált megoldás. A módosítást a K2 csak a legközelebbi indításkor olvassa fel.

5.3. A K2 idejének időszakos beállítása

A K2 is ad egy időpecsétet a kliens oldalról beérkezett ajánlatokra. Mivel a K2 szerver magától nem állítja át a hardver rendszeridejét, célszerű a K2 hardver idejét rendszeresen beállítani a K2 szoftver kézi indítása előtt.

