This packet guide was created by Wyatt, [wyatter@gmail.com](mailto:wyatter@gmail.com), [www.ruosi.org](http://www.ruosi.org), and contains description of all Ultima

Online protocol’s packets since T2A until KR. Last Update Date: 25/11/2007

|  |  |
| --- | --- |
| **00 – Character Creation** | |
| Create a new character | |
|  | |
| byte | ID (00) |
| Integer | 0xedededed |
| Integer | 0xffffffff |
| byte | 0 |
| char[30] | Character Name |
| byte | Gender and Race: (0=human male,1=human female,  2=elf male,3=elf female) |
| byte | Strength |
| byte | Dexterity |
| byte | Intelligence |
| byte | Skill 1 |
| byte | Skill 1 Amount |
| byte | Skill 2 |
| byte | Skill 2 Amount |
| byte | Skill 3 |
| byte | Skill 3 Amount |
| Short | Skin Color |
| Short | Hair Style |
| Short | Hair Color |
| Short | Beard Style |
| Short | Beard Color |
| Short | Starting City |
| Short | 0 |
| Short | Slot |
| Integer | Client IP |
| Short | Shirt color |
| Short | Pants Color |
| **01 - Logout** | |
| Character returns to main menu from character select menu. 5 bytes | |
|  | |
| byte | ID (01) |
| Integer | 0xffffffff |
| **02 - Req Move** | |
| Ask the server if we can walk. 7 bytes | |
|  | |
| byte | ID (02) |
| byte | Direction |
| byte | Sequence Number |
| Integer | Fastwalk Prevention Key |
| **03 - Speech** | |
| Send speech to the server | |
|  | |
| byte | ID (03) |
| Short | Size of Packet |
| byte | Mode (0=say,1=system,2=emote,6=label,7=focus, 8=whipser,9=yell,10=spell,13=guild,14=alliance,15=GM,0xc0=encoded commands) |
| Short | Text Color |
| Short | Font |
| byte[\*] | Text |
| **04 - God Mode Toggle** | |
| Toggles god mode on/off 2 bytes | |
|  | |
| byte | ID (04) |
| byte | God mode on/off |
| **05 – Attack/Last Attack** | |
| Send attack to server 5 bytes | |
|  | |
| byte | ID (05) |
| Integer | Serial of character to attack. If Last Attack and character is null, serial is own serial. |
| **06 - Req Obj Use** | |
| Ask the server if we can use an object (double click) 5 bytes | |
|  | |
| byte | ID (06) |
| Integer | Serial of item double clicked |
| **07 - Req Get Obj** | |
| Ask the server for item 7 bytes | |
|  | |
| byte | ID (07) |
| Integer | Serial of item |
| Short | Amount of item |
| **08 - Req Drop Obj** | |
| Ask the server to drop an item  Grid Index only since 6.0.1.7 2D and 2.45.5.6 KR  0x0F bytes | |
|  | |
| byte | ID (08) |
| Integer | Item Serial |
| Short | X |
| Short | Y |
| сhar  byte | Z  Grid Index |
| Integer | Container Serial (-1 = ground) |
| **09 - Req Look** | |
| Ask the server to look at an item (single click) 5 bytes | |
|  | |
| byte | ID (09) |
| Integer | Serial of item single clicked |
| **0A - Edit** | |
| Edit dynamics and statics 0x0B bytes  Types are: 04 = Dynamic item 07 = NPC 0A = Static Item | |
|  | |
| byte | ID (0A) |
| byte | Type |
| Short | X |
| Short | Y |
| Short | ID |
| char | Z |
| Short | Hue |
| **0B - Edit Area – Old Client** | |
| Edit Area 0x0A bytes | |
|  | |
| byte | ID (0B) |
| byte[9] | Unknown |
| **0B – Damage Packet : since 4.0.7a client** | |
| Damage Packet  0x7 bytes | |
|  | |
| byte | ID (0B) |
| Integer | Serial |
| Short | Amount of damage |
| **0C - Tile Data** | |
| Alter tiledata (send new tiledata to the server) | |
|  | |
| byte | ID (0C) |
| Short | Packet Size |
| Short | Tile ID |
| byte[37] | New Tile data (see tiledata.mul) |
| Short | F777 |
| **0D - NPC Data** | |
| Send new NPC data to the server 3 bytes | |
|  | |
| byte | ID (0D) |
| byte[2] | Unknown |
| **0E - Template Data** | |
| Edit template data | |
|  | |
| Byte | ID (0E) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
| **0F - Paperdoll** | |
| Paperdoll 0x3D bytes | |
|  | |
| Byte | ID (0F) |
| byte[0x3C] | Unknown |
| **10 - Hue Data** | |
| Modify Hue Data 0xD7 bytes | |
|  | |
| Byte | ID (10) |
| byte[0xD6] | Unknown |
| **11 - Mobile Stat** | |
| The status of a character | |
|  | |
| byte | ID (11) |
| Short | Packet Size |
| Integer | Serial |
| char[30] | Character Name |
| Short | Hit Points |
| Short | Max Hit Points |
| byte | Allow Name change (1 = yes,0=no) |
| byte | Supported features from various expansions (sf) |
| byte | Gender |
| Short | Strength |
| Short | Dexterity |
| Short | Intelligence |
| Short | Stamina |
| Short | Max Stamina |
| Short | Mana |
| Short | Max Mana |
| Integer | Gold |
| Short | Armor Rating(Physical Resistance) |
| Short | Weight |
| Short  byte  Short  byte  byte  Short  Short  Short  Short  Short  Short  Short  Integer  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short  Short | Max. Weight ( if sf = 0x5)  Race ( if sf = 0x5)  Stat Cap ( if sf = 0x3)  Followers ( if sf = 0x3)  Max. Followers ( if sf = 0x3)  Fire Resistance ( if sf = 0x4)  Cold Resistance ( if sf = 0x4)  Poison Resistance ( if sf = 0x4)  Energy Resistance ( if sf = 0x4)  Luck ( if sf = 0x4)  Min. Weapon Damage ( if sf = 0x4)  Max. Weapon Damage ( if sf = 0x4)  Tithing Points ( if sf = 0x4)  Hit Chance Increase ( if sf = 0x6)  Swing Speed Increase ( if sf = 0x6)  Damage Chance Increase ( if sf = 0x6)  Lower Reagent Cost ( if sf = 0x6)  Hit Points Regeneration ( if sf = 0x6)  Stamina Regeneration ( if sf = 0x6)  Mana Regeneration ( if sf = 0x6)  Reflect Physical Damage ( if sf = 0x6)  Enhance Potions ( if sf = 0x6)  Defense Chance Increase ( if sf = 0x6)  Spell Damage Increase ( if sf = 0x6)  Faster Cast Recovery ( if sf = 0x6)  Faster Casting ( if sf = 0x6)  Lower Mana Cost ( if sf = 0x6)  Strength Increase ( if sf = 0x6)  Dexterity Increase ( if sf = 0x6)  Intelligence Increase ( if sf = 0x6)  Hit Points Increase ( if sf = 0x6)  Stamina Increase ( if sf = 0x6)  Mana Increase ( if sf = 0x6)  Maximum Hit Points Increase ( if sf = 0x6)  Maximum Stamina Increase ( if sf = 0x6)  Maximum Mana Increase ( if sf = 0x6) |
| **12 - God Command** | |
| External command, use skill, magic, etc  Following Command Types currently not used, it’s from old clients:  0x00 = Go  0x27 = Cast Spell From Spellbook  0x56 = Macro Spell  0x6b = God Mode Command  0xDA = GM Page Query | |
|  | |
| byte | ID (12) |
| Short | Packet Size |
| byte | CommandType |
| byte[\*] | Arguments, may be not present |
| byte | 0 |
| **12.24 – Use Skill/Last Skill** | |
| External command  Use Skill/Last Skill | |
|  | |
| byte | ID (12) |
| Short | Packet Size |
| byte | CommandType(24) |
| byte  byte  byte  byte  byte | 30+Skill ID (ID: from 1 to 55 at present time, if Skill ID = 0, this means that it’s last skill)  Unkown, may be argument, not always presents  20  30  0 |
| **12.43 – Open Spellbook** | |
| External command  Spellbook Type: 31 – Mage Spellbook, 32 – Necromancer Spellbook, 33 – Book Of Chivalry, 34 – Book Of Bushido, 35 – Book Of Ninjitsu, 36 – Spellweaving Spellbook | |
|  | |
| byte | ID (12) |
| Short | Packet Size |
| byte | CommandType(43) |
| byte  byte | Spellbook Type  0 |
| **12.58 – Open Door** | |
| External command  Open Door | |
|  | |
| byte | ID (12) |
| Short | Packet Size |
| byte | CommandType(58) |
| byte | 0 |
| **12.C7 – Action** | |
| External command  Action | |
|  | |
| byte | ID (12) |
| Short | Packet Size |
| byte | CommandType(C7) |
| char[\*]  byte | Action Name  0 |
| **12.F4 – Invoke Virtue** | |
| External command  Invoke Virtue  Virtue ID: 0x1 = Honor Virtue, 0x2= Sacrifice Virtue, 0x3 = Valor Virtue, 0x4 = Compassion Virtue, 0x5 = Honesty Virtue, 0x6 = Humility Virtue, 0x7 = Justice Virtue, 0x8 = Spirituality. | |
|  | |
| byte | ID (12) |
| Short | Packet Size |
| byte | CommandType(F4) |
| Byte  byte | 30+Virtue ID (if Virtue ID = 0, this means none virtue invokes)  0 |
| **13 - Req Obj Equip** | |
| Equip an item 0x0A bytes | |
|  | |
| byte | ID (13) |
| Integer | Item Serial |
| byte | Layer |
| Integer | Container Serial |
| **14 - Elev Change** | |
| Change Z value of item 6 bytes | |
|  | |
| byte | ID (14) |
| Short | X |
| Short  char | Y  Z |
| **15 - Follow** | |
| Follow character 9 bytes | |
|  | |
| byte | ID (15) |
| Integer | Serial1 |
| Integer | Serial2 |
| **16 - Req Script Names** | |
| Get a list of script names 1 byte | |
|  | |
| byte | ID (16) |
| **17 – KR Health Bar Status Update** | |
| KR Health Bar Status Update.  Notes: Server sends this packet both to 2D and KR. When character is poisoned (green health bar), flag is determined as: 0 – remove poison, > 0 – poison level | |
|  | |
| byte | ID (17) |
| byte[2] | Packet Size |
| Integer  Short  Short  Short | Mobile Serial  1  Status Color (1 = Green, 2 = Yellow, others = Red)  Flag (0 = Remove Status Color, 1 = Enable Status Color) |
| **18 - Script Attach** | |
| Add new script to server | |
|  | |
| byte | ID (18) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
| **19 - NPC Convo Data** | |
| Modify NPC Speech data | |
|  | |
| byte | ID (19) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
| **1A – Move** | |
| Move or place an item on the ground | |
|  | |
| byte | ID (1A) |
| Short | Packet Size |
| Integer | Item Serial |
| Short | Item ID |
| Short | Item Amount (if Serial&0x80000000) |
| byte | Stack ID (if Item ID&0x8000) add to Item ID |
| Short | X |
| Short | Y |
| byte | Direction (if X&0x8000) |
| char | Z |
| Short | Hue (if Y&0x8000) |
| byte | Status if (y&0x4000) |
| **1B - Login Confirm** | |
| Player initialization 0x25 bytes | |
|  | |
| byte | ID (1B) |
| Integer | Serial |
| Integer | 0 |
| Short | ID |
| Short | X |
| Short | Y |
| Byte | 0 |
| Char | Z |
| Byte | Direction |
| Integer | 7Fh |
| Integer | 0 |
| Short | 7 |
| Byte | Status |
| Short | Highlight Color |
| Short | 0 |
| Integer | 0 |
| **1C - Text** | |
| Someone is speaking, or the server is sending us info | |
|  | |
| Byte | ID (1C) |
| Short | Packet Size |
| Integer | Character Serial |
| Short | Character ID |
| Byte | Type |
| Short | Text Color |
| Short | Font |
| Char[30] | Name |
| Char[\*] | Text |
| **1D - Destroy Object** | |
| Remove Item from scene 5 bytes | |
|  | |
| Byte | ID (1D) |
| Integer | Item Serial |
| **1E - Animate** | |
| Control Animation 4 bytes | |
|  | |
| Byte | ID (1E) |
| Byte[3] | Unknown |
|  |  |
| **1F - Explode** | |
| Cause explosion 8 bytes | |
|  | |
| Byte | ID (1F) |
| Byte[7] | Unknown |
|  |  |
| **20 - Z Move** | |
| Character is being moved by the server 0x13 bytes | |
|  | |
| Byte | ID (20) |
| Integer | Serial |
| Short | ID |
| Byte | 0 |
| Short | Skin Color |
| Byte | Status |
| Short | X |
| Short | Y |
| Short | 0 |
| Byte | Direction |
| Char | Z |
|  |  |
| **21 - Blocked Move** | |
| Server rejected our walk request 8 bytes | |
|  | |
| byte | ID (21) |
| byte | Sequence Rejected |
| Short | X location to jump back to |
| Short | Y |
| byte | Direction |
| char | Z |
|  |  |
| **22 - OK Move** | |
| Server accepted our walk request 3 bytes | |
|  | |
| byte | ID (22) |
| byte | Sequence accepted |
| byte | Status |
|  |  |
| **23 - Obj Move** | |
| Drag Item 0x1A bytes | |
|  | |
| byte | ID (23) |
| Short | Item ID |
| byte | 0 |
| Short | 0 |
| Short | Item Amount |
| Integer | Source Serial |
| Short | Source X |
| Short | Source Y |
| char | Source Z |
| Integer | Target Serial |
| Short | Target X |
| Short | Target Y |
| char | Target Z |
|  |  |
| **24 - Open Gump** | |
| Open a gump 7 bytes | |
|  | |
| byte | ID (24) |
| Integer | Serial |
| Short | Gump |
| Short | Packet Size |
| Short | Number of Items |
| loop | Item |
| Integer | Item Serial |
| Short | Item ID |
| byte | 0 |
| Short | Item Amount |
| Short | Item X |
| Short | Item Y |
| Integer | Container Serial |
| Short | Item Hue |
| endloop | Item |
|  |  |
| **25 - Obj to Obj** | |
| Add Object to an Object 0x14 bytes | |
|  | |
| byte | ID (25) |
| Integer | Item Serial |
| Short | Item ID |
| byte | 0 |
| Short | Item Amount |
| Short | Item X |
| Short  byte | Item Y  Grid Index (only since 6.0.1.7 2D and 2.45.5.6 KR) |
| Integer | Container Serial |
| Short | Item Color |
|  |  |
| **26 - Old Client** | |
| Old Client, kick 5 bytes | |
|  | |
| byte | ID (26) |
| Integer | Serial of GM who kicked |
|  |  |
| **27 - Get Obj Failed** | |
| Unable to pick up object  2 bytes  Types of reason:  0x0 – cannot lift item,  0x1 – out of range  0x2 – out of sight  0x3 – try to steal  0x4 – are holding  0x5 – inspecific | |
|  | |
| byte | ID (27) |
| byte | Reason of failing get obj |
|  |  |
| **28 - Drop Obj Failed** | |
| Unable to drop object 5 bytes | |
|  | |
| byte | ID (28) |
| Integer | Serial |
|  |  |
| **29 - Drop Obj OK** | |
| Object dropped ok 1 byte | |
|  | |
| byte | ID (29) |
|  |  |
| **2A - Blood** | |
| Blood mode 5 bytes | |
|  | |
| byte | ID (2A) |
| Integer | Serial |
|  |  |
| **2B - God Mode** | |
| God Mode is on/off (server response to packet 04) 2 bytes | |
|  | |
| byte | ID (2B) |
| byte | God mode on/off |
|  |  |
| **2C - Death** | |
| Choose resurrection 2 bytes  Choice = 0=server ask, 1=resurrect, 2=ghost | |
|  | |
| byte | ID (2C) |
| byte | Choice |
|  |  |
| **2D - Health** | |
| Health 0x11 bytes | |
|  | |
| Byte | ID (2D) |
| Integer | Serial |
| Short  Short  Short  Short  Short  Short | Max Hit Points  Hit Points  Max Mana  Mana  Max Stamina  Stamina |
| **2E - Equip Item** | |
| Character is wearing an item 0x0F bytes | |
|  | |
| Byte | ID (2E) |
| Integer | Item Serial |
| Short | Item ID |
| Byte | 0 |
| Byte | Layer |
| Integer | Container Serial |
| Short | Item Color |
|  |  |
| **2F - Swing** | |
| Fight, swing 0x0A bytes | |
|  | |
| Byte | ID (2F) |
| Byte | 0 |
| Integer | Attacker Serial |
| Integer | Defender Serial |
|  |  |
| **30 - Attack OK** | |
| Attack granted 5 bytes | |
|  | |
| Byte | ID (30) |
| Integer | Serial |
|  |  |
| **31 - Attack End** | |
| Attack ended 1 byte | |
|  | |
| byte | ID (31) |
|  |  |
| **32 - Hack Mover** | |
| God mode admin command 2 bytes | |
|  | |
| byte | ID (32) |
| byte | Unknown |
|  |  |
| **33 - Group** | |
| Group command 2 bytes | |
|  | |
| byte | ID (33) |
| byte | Command |
|  |  |
| **34 - Client Query** | |
| Get Status 0x0A bytes  Types:  0x00=God Client  0x04=Basic Status (Packet 0x11)  0x05=Request Skills (Packet 0x3A) | |
|  | |
| byte | ID (34) |
| Integer | 0xedededed |
| byte | Type |
| Integer | Serial |
|  |  |
| **35 - Resource Type** | |
| Get resource type 0x8D bytes | |
|  | |
| byte | ID (35) |
| byte[0x8C] | Unknown |
|  |  |
| **36 - Resource Tile Data** | |
| Resource tile data | |
|  | |
| byte | ID (36) |
| Short | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **37 - Move Object** | |
| Move an object to new location. God Client packet. 8 bytes | |
|  | |
| byte | ID (37) |
| Integer  byte | Item Serial  Z Offset |
| byte  byte | Y Offset  X Offset |
| **38 - Follow Move** | |
| Follow move 7 bytes | |
|  | |
| byte | ID (38) |
| Short  Short  Short | X  Y  Z |
|  |  |
| **39 – Groups** | |
| Groups 9 bytes | |
|  | |
| Byte | ID (39) |
| byte[8] | Unknown |
|  |  |
| **3A - Skills** | |
| Update skills  Client version is only ID, Packet Size, Skill ID, and Lock Status. | |
|  | |
| Byte | ID (3A) |
| Short | Packet Size |
| Byte | List Type (ff=no loop, else loop till ID=0) |
| Loop | Skill |
| Short | Skill ID |
| Short | Skill Value |
| Short | Base Value |
| Byte | Lock Status |
| endloop | Skill |
|  |  |
| **3B - Offer Accept** | |
| Accept Offer | |
|  | |
| byte | ID (3B) |
| Short | Packet Size |
| Integer | Vendor Serial |
| byte | Num of Items (0=clear message) |
| loop | Item |
| byte | Layer |
| Integer | Item Serial |
| Short | Item Amount |
| endloop | Item |
|  |  |
| **3C - Multi Obj to Obj** | |
| Add objects to object | |
|  | |
| byte | ID (3C) |
| Short | Packet Size |
| Short | Number of Items |
| loop | Item |
| Integer | Item Serial |
| Short | Item ID |
| byte | 0 |
| Short | Item Amount |
| Short | Item X |
| Short  byte | Item Y  Grid Index (only since 6.0.1.7 2D and 2.45.5.6 KR) |
| Integer | Container Serial |
| Short | Item Color |
| endloop | Item |
|  |  |
| **3D - Ship** | |
| Ship 2 bytes | |
|  | |
| byte | ID (3D) |
| byte | Unknown |
|  |  |
| **3E - Versions** | |
| Version retrieval 0x25 bytes | |
|  | |
| byte | ID (3E) |
| byte[0x24] | Unknown |
|  |  |
| **3F - Upd Obj Chunk** | |
| Update object chunk | |
|  | |
| byte | ID (3F) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **40 - Upd Terr Chunk** | |
| Update terrain chunk 0xC9 bytes | |
|  | |
| byte | ID (40) |
| byte[0xC8] | Unknown |
|  |  |
| **41 - Update Tile Data** | |
| Send an updated tiledata.mul entry to the client. | |
|  | |
| byte | ID (41) |
| Short | Packet Size |
| Short | Tile ID |
| byte[37] | Tile Data (see tiledata.mul) |
| Short | F777 |
|  |  |
| **42 - Update Art** | |
| Send updated art to client | |
|  | |
| byte | ID (42) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **43 - Update Anim** | |
| Send new animation data to client 0x29 bytes | |
|  | |
| byte | ID (43) |
| byte[0x28] | Unknown |
|  |  |
| **44 - Update Hues** | |
| Send new hue information to the client 0xC9 bytes | |
|  | |
| byte | ID (44) |
| byte[0xC8] | Unknown |
|  |  |
| **45 - Ver Ok** | |
| Ver Ok 5 bytes | |
|  | |
| byte | ID (45) |
| byte[4] | Unknown |
|  |  |
| **46 - New Art** | |
| Send new artwork | |
|  | |
| byte | ID (46) |
| Short | Packet Size |
| Integer | Tile ID |
| byte[\*] | Art Data (see art.mul) |
|  |  |
| **47 - New Terr** | |
| Send new terrain to the server 0x0B bytes | |
|  | |
| byte | ID (47) |
| Short | X |
| Short | Y |
| Short | Art ID |
| Short  Short | Width  Height |
|  |  |
| **48 - New Anim** | |
| Send new animation data (static tile animation) 0x49 bytes | |
|  | |
| byte | ID (48) |
| Integer | Tile ID |
| byte[64] | Frames |
| byte | Unknown |
| byte | Numer of Frames Used |
| byte | Frame Interval |
| byte | Start Interval |
|  |  |
| **49 - New Hues** | |
| Send new hues 0x5D bytes | |
|  | |
| byte | ID (49) |
| Integer | Hue ID |
| Short[32] | Hue Values |
| Short | Start |
| Short | End |
| char[20] | Hue Name |
|  |  |
| **4A - Destroy Art** | |
| Destroy artwork 5 bytes | |
|  | |
| Byte | ID (4A) |
| Integer | Art ID |
|  |  |
| **4B - Check Ver** | |
| Check client version 9 bytes | |
|  | |
| Byte | ID (4B) |
| byte[8] | Unknown |
|  |  |
| **4C - Script Names** | |
| Modify script names | |
|  | |
| byte | ID (4C) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **4D - Script File** | |
| Edit script file | |
|  | |
| byte | ID (4D) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **4E – Set Personal Light Level** | |
| Set the personal light level 6 bytes | |
|  | |
| byte | ID (4E) |
| Integer | Serial |
| char | Level |
|  |  |
| **4F – Set global light level** | |
| Set light level. 2 bytes  0 = Bright, 9 = OSI night, 1F = Black | |
|  | |
| Byte | ID (4F) |
| Char | Level |
|  |  |
| **50 - Board Header – Old Client** | |
| Bulletin Board Header | |
|  | |
| Byte | ID (50) |
| Byte[2] | Packet Size |
| Byte[\*] | Unknown |
|  |  |
| **51 - Board Msg – Old Client** | |
| Bulletin Board Message | |
|  | |
| Byte | ID (51) |
| Byte[2] | Packet Size |
| Byte[\*] | Unknown |
|  |  |
| **52 - Post Msg – Old Client** | |
| Post Bulletin Board Message | |
|  | |
| byte | ID (52) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **53 - Login Reject** | |
| Login Rejected 2 bytes  (1=no character, 2=char exists, 3-4=Can't connect, 5=character already in world,6=login problem,7=idle, 8=can't connect, 9 = character transfer) | |
|  | |
| byte | ID (53) |
| byte | Message |
|  |  |
| **54 - Sound** | |
| Play a sound effect 0x0C bytes  Flags = 0 repeating, 1 = single play | |
|  | |
| byte | ID (54) |
| byte | Flags |
| Short | Effect |
| Short | Volume |
| Short | X |
| Short | Y |
| Short | Z |
|  |  |
| **55 - Login Complete** | |
| Begin Game 1 byte | |
|  | |
| byte | ID (55) |
|  |  |
| **56 - Map Command** | |
| Plot course for ships 0x0B bytes | |
|  | |
| byte | ID (56) |
| Integer | Serial |
| byte | Action (1=add,2=insert,3=change,4=remove,5=clear,6=toggle edit) |
| byte | Pin Number |
| Short | Pin X |
| Short | Pin Y |
|  |  |
| **57 - Upd Regions** | |
| Update regions 0x6E bytes | |
|  | |
| byte | ID (57) |
| byte[0x6D] | Unknown |
|  |  |
| **58 - New Region** | |
| Create a new region 0x6A bytes | |
|  | |
| byte | ID (58) |
| char[40] | Area Name |
| Integer | 0 |
| Short | X |
| Short | Y |
| Short | Width |
| Short | Height |
| Short | Z1 |
| Short | Z2 |
| char[40] | Description |
| Short | Sound FX |
| Short | Music |
| Short | Night Sound FX |
| Byte | Dungeon |
| Short | Light |
|  |  |
| **59 - New Context FX** | |
| Create a new effect | |
|  | |
| Byte | ID (59) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **5A - Upd Context FX** | |
| Update effect | |
|  | |
| byte | ID (5A) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **5B - Game Time** | |
| Set the time of day 4 bytes | |
|  | |
| byte | ID (5B) |
| byte | Hours |
| byte | Minutes |
| byte | Seconds |
|  |  |
| **5C - Restart Ver** | |
| Resart Ver 2 bytes | |
|  | |
| byte | ID (5C) |
| byte | Unknown |
|  |  |
| **5D - Pre Login** | |
| Select the character to play 0x49 bytes | |
|  | |
| byte | ID (5D) |
| Integer | 0xEDEDEDED |
| char[32] | Character Name |
| Integer[7] | Unknown |
| Integer | Character Index |
| Integer | Client IP |
|  |  |
| **5E - Server List** | |
| Server List | |
|  | |
| byte | ID (5E) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **5F - Server Add** | |
| Add server 0x31 bytes | |
|  | |
| byte | ID (5F) |
| byte[0x30] | Unknown |
|  |  |
| **60 - Server Remove** | |
| Delete a server 5 bytes | |
|  | |
| Byte | ID (60) |
| byte[4] | Unknown |
|  |  |
| **61 - Destroy Static** | |
| Delete a static. God Client Packet. 9 bytes | |
|  | |
| Byte | ID (61) |
| Short | X |
| Short | Y |
| Short | Z |
| Short | ID |
|  |  |
| **62 - Move Static** | |
| Move a static. God Client packet.  0x0F bytes | |
|  | |
| Byte | ID (62) |
| Short  Short  Short  Short  Short  Short  Short | Old X  Old Y  Old Z  Item ID  Z Offset  Y Offset  X Offset |
|  |  |
| **63 - Area Load** | |
| Load an area 0x0D bytes | |
|  | |
| Byte | ID (63) |
| byte[0x0C] | Unknown |
|  |  |
| **64 - Area Load Req** | |
| Attempt to load area 1 byte | |
|  | |
| byte | ID (64) |
|  |  |
| **65 - Weather Change** | |
| Set current weather 4 bytes  Type: 0x00 = "It starts to rain" 0x01 = "A fierce storm approaches" 0x02 = "It begins to snow" 0x03 = "A storm is brewing." 0xfe = set temperature 0xff = Stop all weather. | |
|  | |
| byte | ID (65) |
| byte | Weather Type |
| byte | Number of weather effects |
| byte | Temperature |
|  |  |
| **66 - Book Page** | |
| Show book As a client message, this writes to the book. | |
|  | |
| byte | ID (66) |
| Short | Packet Size |
| Integer | Item Serial |
| Short | Number of Pages |
| loop | Page |
| Short | Page Index |
| Short | Number of Lines |
| loop | Line |
| byte[\*] | Text |
| endloop | Line |
| endloop | Page |
|  |  |
| **67 – Simped** | |
| Simped 0x15 bytes | |
|  | |
| byte | ID (67) |
| byte[0x14] | Unknown |
|  |  |
| **68 - Script LS Attach** | |
| Add LS script | |
|  | |
| Byte | ID (68) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **69 - Friends** | |
| Unknown | |
|  | |
| Byte | ID (69) |
| Short | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **6A - Friend Notify** | |
| Notify Friend 3 bytes | |
|  | |
| Byte | ID (6A) |
| byte[2] | Unknown |
|  |  |
| **6B - Key Use** | |
| Use Key 9 bytes | |
|  | |
| Byte | ID (6B) |
| byte[8] | Unknown |
|  |  |
| **6C - Target** | |
| Bring up targeting cursor 0x13 bytes | |
|  | |
| Byte | ID (6C) |
| Byte | Target Type |
| Integer | Character Serial |
| Byte | Check Crime |
| Integer | Item Serial |
| Short | X (ffff=cancel) |
| Short | Y (ffff=cancel) |
| Short | Z |
| Short | Graphic |
|  |  |
| **6D - Music** | |
| Play a midi music file 3 bytes | |
|  | |
| Byte | ID (6D) |
| Short | Midi Song |
|  |  |
| **6E - Anim** | |
| Show an animation 0x0E bytes | |
|  | |
| byte | ID (6E) |
| Integer | Serial |
| Short | Action |
| Short | Frame Count |
| Short | Repeat Times |
| byte | Forward (0=forward 1= backward) |
| byte | Repeat Flag |
| byte | Delay (0=fast, 0xff=slow) |
|  |  |
|  |  |
| **6F - Trade** | |
| Open trade window  Type:  0 = Start Trading  1 = Cancel Trade  2 = Reset Checkmarks | |
|  | |
| byte | ID (6F) |
| Short | Packet Size |
| byte | Type (0=start,1=cancel,2=change checkmarks) |
| Integer | Character Serial |
| Integer | Item 1 Serial (type0 only) |
| Integer | Item 2 Serial (type0 only) |
| byte | Name Included (type 0 only) |
| char[30] | Name (type 0 only, only if Name Included) |
|  |  |
| **70 – Effect** | |
| Play a special effect (like magic) 0x1C bytes  Type:  00 = Go from source to destination  01 = Lightning strike  02 = Stay at location  03 = Stay with source | |
|  | |
| byte | ID (70) |
| byte | Type |
| Integer | Character Serial |
| Integer | Target Serial |
| Short | Object ID |
| Short | X |
| Short | Y |
| char | Z |
| Short | Target X |
| Short | Target Y |
| char | Target Z |
| byte | Speed |
| byte  Short  byte | Duration  0  Fixed Duration |
| byte | Explode |
|  |  |
| **71 - BBoard** | |
| Display a message board  Type:  0=draw board,  1=get posts,  2-3=get body,  4=ack dl,  5=post,  6=delete | |
|  | |
| Byte | ID (71) |
| Short | Packet Size |
| Byte | Type |
|  |  |
| **71.0 - Draw Board** | |
| Draw the Bulletin board | |
|  | |
| Byte | ID (71) |
| Short | Packet Size |
| Byte | Type (0) |
| Integer | Board Serial |
| char[22] | Board Name |
| Integer | ID |
| Integer | 0 |
|  |  |
| **71.1 - Message List** | |
| List of bulletin board messages | |
|  | |
| Byte | ID (71) |
| Short | Packet Size |
| Byte | Type (1) |
| Integer | Board Serial |
| Integer | Message Serial |
| Integer | Container Serial (0 = top level) |
| Byte | Name Length |
| char[\*] | Name |
| Byte | Subject Length |
| char[\*] | Subject |
| Byte | Time Length |
| char[\*] | Time |
|  |  |
| **71.2 - Message** | |
| A single message on a bulletin board | |
|  | |
| Byte | ID (71) |
| Short | Packet Length |
| Byte | Type (2) |
| Integer | Board Serial |
| Integer | Message Serial |
| byte | Name Length |
| char[\*] | Name |
| byte | Subject Length |
| char[\*] | Subject |
| byte | Time Length |
| char[\*] | Time |
| byte[29] | Unknown |
| byte | Num Lines |
| loop | Line |
| byte | Line Length |
| char[\*] | Line Text |
| endloop | Line |
|  |  |
| **71.3 - Get Message** | |
| Ask the server for a message | |
|  | |
| byte | ID (71) |
| Short | Packet Size |
| byte | Type (3) |
| Integer | Board Serial |
| Integer | Message Serial |
|  |  |
| **71.4 - Get Summary** | |
| Get a bulletin board's message summary | |
|  | |
| byte | ID (71) |
| Short | Packet Size |
| byte | Type (4) |
| Integer | Board Serial |
| Integer | Message Serial |
|  |  |
| **71.5 - Post Message** | |
| Post a message to a bulletin board | |
|  | |
| byte | ID (71) |
| Short | Packet Size |
| byte | Type (5) |
| Integer | Board Serial |
| Integer | Reply Message Serial (0 if no reply) |
| byte | Subject Length |
| char[\*] | Subject |
| byte | Num Lines |
| loop | Line |
| byte | Line Length |
| char[\*] | Line Text |
| endloop | Line |
|  |  |
| **71.6 - Delete Message** | |
| Delete a posted message | |
|  | |
| Byte | ID (71) |
| Short | Packet Size |
| Byte | Type (6) |
| Integer | Board Serial |
| Integer | Message Serial |
|  |  |
| **72 - Combat** | |
| Set war mode and stuff 5 bytes | |
|  | |
| Byte | ID (72) |
| Byte | War Mode |
| Byte | 0 |
| Byte  byte | 0x32  0 |
|  |  |
| **73 - Ping** | |
| Ping and Pong 2 bytes | |
|  | |
| Byte | ID (73) |
| Byte | Value |
|  |  |
| **74 - Shop Data** | |
| Purchase details | |
|  | |
| Byte | ID (74) |
| Short | Packet Size |
| Integer | Vendor Serial |
| Byte | Number of Items |
| Loop | Item |
| Integer | Price |
| byte | Name Length |
| char[\*] | Name |
| endloop | Item |
|  |  |
| **75 - Rename MOB** | |
| Rename character 0x23 bytes | |
|  | |
| byte | ID (75) |
| Integer | Serial |
| char[30] | Name |
|  |  |
| **76 - Server Change** | |
| Server change 0x10 bytes | |
|  | |
| byte | ID (76) |
| Short  Short  Short  byte  Short  Short  Short  Short | X  Y  Z  0  0  0  Width  Height |
|  |  |
| **77 - Naked MOB** | |
| Another character or monster is walking (why is this called Naked MOB?) 0x11 bytes | |
|  | |
| Byte | ID (77) |
| Integer | Serial |
| Short | ID |
| Short | X |
| Short | Y |
| Char | Z |
| Byte | Direction |
| Short | Skin Color |
| Byte | Status |
| Byte | Notoriety (Murderer, Friend etc) |
|  |  |
| **78 - Equipped MOB** | |
| Add a character to the scene | |
|  | |
| Byte | ID (78) |
| Short | Packet Size |
| Integer | Serial |
| Short | ID |
| Short | Amount/Corpse (if Serial&0x80000000) |
| Short | X |
| Short | Y |
| Char | Z |
| Byte | Direction |
| Short | Skin Color |
| Byte | Status |
| Byte | Notoriety |
| Loop | Item (while Serial!=0) |
| Integer | Item Serial |
| Short | Item ID |
| Byte | Item Layer |
| Short | Item Color (if Item ID&0x8000) |
| endloop  Integer | Item  0 |
|  |  |
| **79 - Resource Query** | |
| Get Resource 9 bytes | |
|  | |
| byte | ID (79) |
| Integer | Unknown |
| Integer | Unknown |
|  |  |
| **7A - Resource Data** | |
| Resource Data | |
|  | |
| byte | ID (7A) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **7B - Sequence** | |
| Sequence 2 bytes | |
|  | |
| byte | ID (7B) |
| byte | Key |
|  |  |
| **7C - Obj Picker** | |
| Display a Gump with options (like the Add menu) | |
|  | |
| byte | ID (7C) |
| Short | Packet Size |
| Integer | Gump Serial |
| Short | Gump ID |
| byte | Title Length |
| byte[\*] | Title |
| byte | Number of Lines |
| loop | Line |
| Short | Choice ID |
| Short | Hue |
| byte | Line Length |
| char[\*] | Text |
| endloop | Line |
|  |  |
| **7D - Picked Obj** | |
| Choose an option from 7C 0x0D bytes | |
|  | |
| byte | ID (7D) |
| Integer | Gump Serial |
| Short  Short | Gump ID  Index |
| Short | Item ID |
| Short | Hue |
|  |  |
| **7E - God View Query** | |
| Get god view data 2 bytes | |
|  | |
| byte | ID (7E) |
| byte | God view stat id |
|  |  |
| **7F - God View Data** | |
| God view data | |
|  | |
| byte | ID (7F) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **80 - Acct Login Req** | |
| Login to the login server. 0x3E bytes | |
|  | |
| byte | ID (80) |
| char[30] | Name |
| char[30] | Password |
| byte | Unknown |
|  |  |
| **81 - Acct Login Ok** | |
| Login ok  5+(Max Characters Per Account)\*60 bytes | |
|  | |
| byte | ID (81) |
| byte[2] | Packet Size |
| byte | Number of Characters |
| byte  loop  byte[30]  byte[30] | 0  Max Number of Characters  Name  Password |
| **82 - Acct Login Fail** | |
| There was an error logging in 2 bytes  0=No Account, 1=Account in Use, 2=Account Blocked, 3=No Password, 6 = IGR, 9 = Character Transfer, 0xFE=Idle, 0xFF=Bad Communication | |
|  | |
| byte | ID (82) |
| byte | Error |
|  |  |
| **83 - Acct Del Char** | |
| Delete a player 0x27 bytes | |
|  | |
| byte | ID (83) |
| byte[30] | Password |
| Integer | Character Index |
| Integer | Client IP |
|  |  |
| **84 - Chg Char PW** | |
| Change Password 0x45 bytes | |
|  | |
| Byte | ID (84) |
| byte[0x44] | Unknown |
|  |  |
| **85 - Chg Char Result** | |
| Result of Change character request 2 bytes  Result Types:  0=Invalid Password,1=character not exist,2=character in game,3=character too young,4=character in queue,5=bad result | |
|  | |
| Byte | ID (85) |
| Byte | Result Type |
|  |  |
| **86 - All Characters** | |
| Resend Characters after delete | |
|  | |
| Byte | ID (86) |
| Short | Packet Size |
| Byte | Number of Characters |
| Loop | Character |
| char[30] | Name |
| char[30] | Password |
| Endloop | Character |
|  |  |
| **87 - Send Resources** | |
| Send resources | |
|  | |
| Byte | ID (87) |
| Short | Packet Size |
| Short | 0x101 |
| Short | 0 |
| Byte | Unknown |
| Integer | X |
| Integer | Y |
| Integer | X2 |
| Integer | Y2 |
| byte | Name Length |
| char[\*] | Name |
|  |  |
| **88 - Open Paper Doll** | |
| Display paper doll 0x42 bytes | |
|  | |
| byte | ID (88) |
| Integer | Character Serial |
| char[60] | Name |
| byte | Status |
|  |  |
| **89 - Corpse EQ** | |
| Corpses backpack | |
|  | |
| byte | ID (89) |
| Short | Packet Size |
| Integer | Container Serial |
| loop | Item (while Layer!=0) |
| byte | Layer |
| Integer | Item Serial |
| endloop | Item |
| byte | 0 |
| **8A - Trigger Edit** | |
| Edit Trigger | |
|  | |
| byte | ID (8A) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **8B - Display Sign** | |
| Show Sign | |
|  | |
| byte | ID (8B) |
| byte[2]  Integer | Packet Size  Serial |
| Short | Gump ID |
| Short  byte[\*]  Short  byte[\*] | Text Length  Text  Caption Length  Caption |
| **8C - User Server** | |
| Relay to the game server. 0x0B bytes | |
|  | |
| byte | ID (8C) |
| Integer | Server IP |
| Short | Server Port |
| Integer | Auth ID |
|  |  |
| **8D – KR Character Creation** | |
| Create a new character in KR client, 146 bytes  Note: Client Flags = 0x41. It’s wrong flags value, must be 0x3F, possibly we need to use offset value from 0xE1 packet. | |
|  | |
| byte  Short | ID (00)  Packet Size |
| Integer | 0xedededed |
| Integer | Character Index |
| char[30]  char[30] | Character Name  “Unknown” |
| byte  byte  byte  byte | Profession  Client Flags  Gender (male = 0, female = 1)  Race (human = 0, elf = 1) |
| byte | Strength |
| byte | Dexterity |
| byte | Intelligence |
| Short  Integer  Integer | Skin Color  0  0 |
| byte | Skill 1 Amount |
| byte | Skill 2 |
| byte | Skill 2 Amount |
| byte | Skill 3 |
| byte | Skill 3 Amount |
| byte  byte  byte[25]  byte | Skill 4  Skill 4 Amount  0  0x0B |
| Short | Hair Style |
| Short  byte  Integer  byte  Short  Short  byte  Short  Short  byte | Hair Color  0x0C  0  0x0D  Shirt Color  Shirt ItemID  0x0F  Face Color  Face ID  0x10 |
| Short | Beard Style |
| Short | Beard Color |
| **8E - Move Character** | |
| Move Character | |
|  | |
| byte | ID (8E) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **8F - Unused4** | |
| Unused | |
|  | |
| byte | ID (8F) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **90 - Open Course Gump** | |
| Show map, for plotting etc. 0x13 bytes | |
|  | |
| byte | ID (90) |
| Integer | Serial |
| Short | Corner Image |
| Short | X1 |
| Short | Y1 |
| Short | X2 |
| Short | Y2 |
| Short | Width |
| Short | Height |
|  |  |
| **91 - Post Login** | |
| Login to the game server 0x41 bytes | |
|  | |
| byte | ID (91) |
| Integer | Auth ID |
| char[30] | Name |
| char[30] | Password |
|  |  |
| **92 - Upd Multi** | |
| Update Multi | |
|  | |
| byte | ID (92) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **93 - Book Header – Old Client** | |
| Open a book 0x63 bytes | |
|  | |
| byte | ID (93) |
| Integer | Item Serial |
| byte | Editable |
| byte | 1 |
| Short | Number of Pages |
| char[60] | Title |
| char[30] | Author |
|  |  |
| **94 - Upd Skill** | |
| Alter skill | |
|  | |
| Byte | ID (94) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **95 - Hue Picker** | |
| Dye 9 bytes | |
|  | |
| Byte | ID (95) |
| Integer | Item Serial |
| Short | Item ID |
| Short | Color |
|  |  |
| **96 - Game Cent Mon** | |
| Monitor game, mostly through godview | |
|  | |
| Byte | ID (96) |
| Short | Packet Size |
| Byte | Command |
| byte[\*] | Unknown |
|  |  |
| **97 - Player Move** | |
| Player move 2 bytes | |
|  | |
| Byte | ID (97) |
| Byte | Direction |
|  |  |
| **98 - MOB Name** | |
| Alter name  37 bytes | |
|  | |
| Byte | ID (98) |
| byte[2]  Integer | Packet Size  Serial |
| byte[30] | Name |
|  |  |
| **99 - Target Multi** | |
| Targeting cursor for multis 0x1A bytes | |
|  | |
| Byte | ID (99) |
| Byte | Allow Ground |
| Integer  Byte | Deed Serial  Status |
| Byte[11] | 0 |
| Short | Multi ID |
| Short | X |
| Short | Y |
| Short | Z |
|  |  |
|  |  |
| **9A - Text Entry** | |
| Text Entry | |
|  | |
| byte | ID (9A) |
| byte[2] | Packet Size |
| Integer  Integer | Serial  Prompt |
| Integer  byte[\*] | Type  Text |
| **9B - Request Assist** | |
| Page a GM 2 bytes | |
|  | |
| Byte | ID (9B) |
| Byte | Unknown |
|  |  |
| **9C - Assist Request** | |
| Assistance response 0x35 bytes | |
|  | |
| Byte | ID (9C) |
| byte[0x34] | Unknown |
|  |  |
| **9D - GM Single** | |
| GM Single 0x33 bytes | |
|  | |
| Byte | ID (9D) |
| byte[0x32] | Unknown |
|  |  |
| **9E - Shop Sell** | |
| Details for sales | |
|  | |
| Byte | ID (9E) |
| Short | Packet Size |
| Integer | Vendor Serial |
| Short | Number of Items |
| loop | Item |
| Integer | Item Serial |
| Short | Item ID |
| Short | Item Color |
| Short | Item Amount |
| Short | Value |
| Short | Name Length |
| char[\*] | Name |
| endloop | Item |
|  |  |
| **9F - Shop Offer** | |
| Sell window | |
|  | |
| byte | ID (9F) |
| Short | Packet Size |
| Integer | Vendor Serial |
| Short | Number of Items |
| loop | Item |
| Integer | Item Serial |
| Short | Item Amount |
| endloop | Item |
|  |  |
|  |  |
| **A0 - Britannia Select** | |
| Select a shard to play on 3 bytes | |
|  | |
| byte | ID (A0) |
| Short | Server Index |
|  |  |
| **A1 - HP Health** | |
| Adjust strength and Hit Points 9 bytes | |
|  | |
| byte | ID (A1) |
| Integer | Serial |
| Short | Max Hit Points |
| Short | Hit Points |
|  |  |
| **A2 - Mana Health** | |
| Update Intelligence and Mana 9 bytes | |
|  | |
| byte | ID (A2) |
| Integer | Serial |
| Short | Max Mana |
| Short | Mana |
|  |  |
| **A3 - Fat Health** | |
| Update Dexterity and Stamina 9 bytes | |
|  | |
| byte | ID (A3) |
| Integer | Serial |
| Short | Max Stamina |
| Short | Stamina |
|  |  |
| **A4 - Hardware Info** | |
| Send system info to the server 0x95 bytes | |
|  | |
| byte | ID (A4) |
| byte[0x94] | Unknown |
|  |  |
| **A5 - Web Browse** | |
| Launch a browser and to URL | |
|  | |
| byte | ID (A5) |
| Short | Packet Size |
| char[\*] | URL |
|  |  |
| **A6 - Message** | |
| Show a scroll on the screen | |
|  | |
| byte | ID (A6) |
| Short | Packet Size |
| byte | Font |
| Integer | Tip Number |
| Short | Text Length |
| char[\*] | Text |
|  |  |
|  |  |
| **A7 - Req Tip** | |
| Get a tooltip 4 bytes | |
|  | |
| Byte | ID (A7) |
| Short | Tip |
| byte | 0 = tip, 1 = notice |
|  |  |
| **A8 - Britannia List** | |
| List the available shards | |
|  | |
| byte | ID (A8) |
| Short | Packet Size |
| byte | FFh |
| Short | Number of Servers |
| loop | Server |
| Short | Server Index |
| char[32] | Server Name |
| byte | Percent Full |
| byte | Timezone |
| Integer | Server IP |
| endloop | Server |
|  |  |
| **A9 - Cities and Chars** | |
| List of characters as well as starting cities  Flags (each flag is for each feature, if you need to combine features, you need to summ flags): 0x2 = overwrite configuration button; 0x4 = limit 1 character per account; 0x8 = enable context menus; 0x10 = limit character slots; 0x20 = paladin and necromancer classes; 0x40 = 6th character slot; 0x80 = samurai and ninja classes; 0x100 = elven race; 0x200 = KR support flag1; 0x400 = KR support flag2; 0x1000 = 7th character slot – only 2D client | |
|  | |
| byte | ID (A9) |
| Short | Packet Size |
| byte | Number of Characters |
| loop | Character |
| char[30]  char[30] | Character Name  Password |
| endloop | Character |
| Byte | Number of Cities |
| Loop | City |
| Byte | Index |
| char[31] | City Name |
| char[31] | Area of Town |
| endloop | City |
| Integer | Flags (May not be present) |
|  |  |
| **AA - Current Target** | |
| Current attack target 5 bytes | |
|  | |
| byte | ID (AA) |
| Integer | Serial (0 = attack invalid) |
|  |  |
| **AB - String Query** | |
| Text Entry | |
|  | |
| byte | ID (AB) |
| Short | Packet Size |
| Integer | Serial |
| byte | Parent ID |
| byte | Buttom ID |
| Short | Text Length |
| char[\*] | Text |
| byte | Style (0=none, 1=normal, 2=numerical) |
| Integer | Max Length |
| Short | Label Length |
| char[\*] | Label |
|  |  |
| **AC - String Response** | |
| Text Entry Response | |
|  | |
| byte | ID (AC) |
| Short | Packet Size |
| Integer | Serial |
| byte | Type |
| byte | Index |
| byte | Mode (0=cancel,1=ok) |
| Short | Text Length |
| char[\*] | Text |
|  |  |
| **AD - Speech Unicode** | |
| Speak in unicode  If Mode&0xc0 then there are keyShorts (from speech.mul) present.  KeyShorts are using in UO since 2.0.7 client  KeyShorts: The first 12 bits = the number of keyShorts present. The keyShorts are included right after this, each one is 12 bits also.  The keyShorts are padded to the closest byte. For example, if there are 2 keyShorts, it will take up 5 bytes. 12bits for the number, and 12 bits for each keyShort. 12+12+12=36. Which will be padded 4 bits to 40 bits or 5 bytes. | |
|  | |
| Byte | ID (AD) |
| Short | Packet Length |
| Byte | Mode(0=say,2=emote,8=whisper,9=yell,10=spell,13=guild,14=alliance,15=GM,0xc0=encoded commands) |
| Short | Text Color |
| Short | Font |
| char[4] | Language |
| byte[\*] | KeyShorts (Only present if mode&0xc0) |
| char[\*] | Text |
|  |  |
| **AE - Text Unicode** | |
| Someone is speaking in Unicode | |
|  | |
| Byte | ID (AE) |
| Short | Packet Size |
| Integer | Serial |
| Short | ID |
| Byte | Type |
| Short | Text Color |
| Short | Font |
| char[4] | Language |
| char[30] | Name |
| char[\*] | Text |
|  |  |
| **AF - Death Anim** | |
| Death Anim 0x0D bytes | |
|  | |
| byte | ID (AF) |
| Integer | Character Serial |
| Integer | Corpse Serial |
| Integer | 0 |
|  |  |
| **B0 - Generic Gump – Old Client** | |
| Show Gump | |
|  | |
| byte | ID (B0) |
| Short | Packet Size |
| Integer | Serial |
| Integer | Gump ID |
| Integer | X |
| Integer | Y |
| Short | Layout Length |
| char[\*] | Layout |
| Short | Number of Lines |
| loop | Line |
| Short | Line Length |
| Short[\*] | Text |
| endloop | Line |
|  |  |
| **B1 - Gen Gump Trig** | |
| Generic Gump Choice | |
|  | |
| byte | ID (B1) |
| Short  Integer  Integer  Integer  Integer  loop  Integer  endloop  Integer  loop  Short  Short  byte[\*]  endloop  Integer  Integer | Packet Size  Gump Serial  Gump ID  Button ID  Switches Count  Switch  Switch ID  Switch  Text Entry Count  Text Entry  Text Entry ID  Text Entry Length  Text Entry Text  Text Entry  Switches Count (Only if Gump ID = 461)  Beheld Serial (Only if (Gump ID = 461 && Button ID = 1 && Switches Count > 0)) |
|  |  |
|  |  |
| **B2 - Chat Msg** | |
| Chat Message | |
|  | |
| byte | ID (B2) |
| byte[2]  Short  char[4] | Packet Size  Message Number  Language |
| char[\*]  char[\*] | Param1  Param2 |
|  |  |
| **B3 - Chat Text** | |
| Chat Text | |
|  | |
| byte | ID (B3) |
| byte[2]  char[4]  Short | Packet Size  Language  Action |
| byte[\*] | Parameters |
|  |  |
| **B4 - Target Obj List** | |
| Target object list | |
|  | |
| byte | ID (B4) |
| byte[2] | Packet Size |
| byte[\*] | Unknown |
|  |  |
| **B5 - Chat Open** | |
| Chat open 0x40 bytes | |
|  | |
| byte | ID (B5) |
| byte[0x39] | Chat Name |
|  |  |
| **B6 - Help Request** | |
| Show Popup Help 9 bytes | |
|  | |
| byte | ID (B6) |
| Integer | Help Serial |
| byte | Langauge Number |
| char[3] | Language |
|  |  |
| **B7 - Help Unicode Text** | |
| Display Popup help | |
|  | |
| byte | ID (B7) |
| Short | Packet Size |
| Integer | Help Serial |
| Short[\*] | Unicode Text |
|  |  |
| **B8 - Char Profile** | |
| Profile(Client version)  Mode:  0x00=display request,0x01=edit request, | |
|  | |
| byte | ID (B8) |
| Short | Packet Size |
| byte | Mode |
| Integer  Short  Short | Serial  Unknown ( if Mode = 0x01 )  Length ( if Mode = 0x01 ) |
| byte[\*] | Unicode text ( if Mode = 0x01 ) |
| **B8 - Char Profile** | |
| Profile(Server version)  Mode:  0x00=display request,0x01=edit request, | |
|  | |
| byte | ID (B8) |
| Short | Packet Size |
| Integer | Serial |
| byte[\*]  byte[\*]  byte[\*] | Header  Body  Footer |
|  |  |
| **B9 - Features** | |
| Expansions Features Enabling  3 bytes  Flags (each flag is for each feature, if you need to combine features, you need to summ flags):  0x01 = enable T2A features: chat button, regions; 0x02 = enable renaissance features; 0x04 = enable third down features; 0x08 = enable LBR features: skills, map; 0x10 = enable AOS features: skills, spells, map, fightbook; 0x20 = enable 6th character slot; 0x40 = enable SE features: spells, skills, map; 0x80 = enable ML features: elven race, spells, skills; 0x100 = enable The Eight Age splash screen; 0x200 = enable The Ninth Age splash screen; 0x1000 = enable 7th character slot; 0x2000 = enable The Tenth Age KR faces  Note1: this packet is send immediately after login. Note2: on OSI  servers this controls features: OSI enables/disables it via “upgrade codes.” | |
|  | |
| byte | ID (B9) |
| Short | Flags |
| **BA - Pointer** | |
| Display Quest Pointer 6 bytes | |
|  | |
| byte | ID (BA) |
| byte | Active |
| Short | X |
| Short | Y |
| **BB – Ultima Messenger** | |
| Ultima Messenger 9 bytes | |
|  | |
| byte | ID (BB) |
| byte[8] | Unknown |
| **BC - Game Season** | |
| Season 3 bytes | |
|  | |
| byte | ID (BC) |
| byte | Season (0=spring, 1=summer, 2=fall, 3=winter, 4 = desolation) |
| byte | Play Sound( 1=yes, 0=no) |
| **BD – Client/Server Version** | |
| Send client version to the server.  Note: This packet works only in 2D client. KR client hasn’t any version’s determination algorithm. | |
|  | |
| byte | ID (BD) |
| Short | Packet Size |
| char[\*] | Version String |
|  |  |
| **BE - Assist Version** | |
| Assist Version | |
|  | |
| Byte | ID (BE) |
| byte[2]  Integer | Packet Size  Serial |
| byte[\*] | Version String |
|  |  |
| **BF - Generic Command** | |
| Miscellaneous Commands | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command |
|  |  |
| **BF.1 - Fast Walk** | |
| Cycle's through the keys in the stack when walking. | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (1) |
| Integer[6] | Key Stack |
|  |  |
| **BF.10 – OPLInfo – Old Client** | |
| OPLInfo packet. | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (10) |
| Integer  Integer | Serial  Hash |
| **BF.10 – Display Equipment Info** | |
| Display Equipment Info (Server version)  Client version is Query Properties and has only Serial. | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (10) |
| Integer  Integer  Integer  Short  char[\*]  Integer  loop  Integer  Short  endloop  Integer | Serial  Info Number  -3 ( if Owner of equipment != null )  Owner Name Length ( if Owner of equipment != null )  Owner Name ( if Owner of equipment != null )  -4 ( if equipment is unidentified )  Attribute  Number  Charges  Attribute  -1 |
| **BF.13 – Context Menu Request** | |
| Context Menu Request | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (13) |
| Integer | Serial |
|  |  |
| **BF.14 – Display Context Menu in 2D client** | |
| Display Context Menu  Flags: 0x00 = Enabled, 0x01 = Disabled, 0x20 = Colored | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (14) |
| Short  Integer  byte  loop  Short  Short  Short  Short  endloop | 1  Serial  Length  Context Menu Entry  Index of Entry  Number  Flags  Hue (if Flags = 0x20)  Context Menu Entry |
| **BF.14 – Display Context Menu in KR client** | |
| Display Context Menu  Flags: 0x00 = Enabled, 0x01 = Disabled, 0x04 = Highlighted | |
|  | |
| Byte | ID (BF) |
| Short | Size |
| Short | Command (14) |
| Short  Integer  byte  loop  Integer  Short  Short  endloop | 2  Serial  Length  Context Menu Entry  Number  Index of Entry  Flags  Context Menu Entry |
| **BF.15 – Context Menu Response** | |
| Context Menu Response | |
|  | |
| Byte | ID (BF) |
| Short | Size |
| Short | Command (15) |
| Integer  Short | Serial  Index (if Serial != 0) |
| **BF.17 – Display Help Topic** | |
| Display Help Topic | |
|  | |
| Byte | ID (BF) |
| Short | Size |
| Short | Command (17) |
| byte  Integer  byte | 1  Topic ID  Display (1=yes,0=no) |
| **BF.18 - Enable Map Diffs** | |
| This packet is sent by the server to the client, telling the client to use the mapdif\* and stadif\* files to patch the map and statics.  Usually there are 5 maps in this packet, 0 = fellucca, 1 = trammel, 2 = ilshenar, 3 = malas, and 4 = tokuno | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (18) |
| Integer | Number of Maps |
| loop[map] |  |
| Integer | Number of Map patches in this map |
| Integer | Number of Static patches in this map |
| endloop[map] |  |
|  |  |
| **BF.19 – Miscellaneous status** | |
| Miscellaneous status is BF Command 19 | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (19) |
| byte | Party Command |
| **BF.19.0 – Bonded Status** | |
| Bonded Status | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (19) |
| byte | Subcommand(0) |
| Integer | Serial |
| byte | 1 |
|  |  |
| **BF.19.2 – Stat Lock Info** | |
| Stat Lock Info | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (19) |
| byte | Subcommand(2) |
| Integer | Serial |
| byte | 0 |
| byte | Lock Flags |
|  |  |
| **BF.19.5 – Stat Lock Info KR** | |
| Stat Lock Info KR | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (19) |
| Byte | Subcommand(5) |
| Integer | Serial |
| Byte | 0 |
| Byte | Lock Flags |
| byte  Integer | 0  0 |
| **BF.19.5 –Update Mobile Status Animation** | |
| Update Mobile Status Animation (Uses for character statues) | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (19) |
| Byte | Subcommand(5) |
| Integer | Serial |
| Byte | 0 |
| Byte | FF |
| Byte  Byte  Byte  Byte  byte | Status  0  Animation  0  Frame |
| **BF.1A – Stat Lock Change** | |
| Stat Lock Change  Stat Type: 0x0 = strength, 0x1 = dexterity, 0x2 = intelligence Lock Value: 0x0 = up, 0x1 = down, 0x2 = locked | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (1A) |
| byte | Stat Type |
| byte | Lock Value |
|  |  |
|  |  |
| **BF.1B – New Spellbook Content** | |
| New Spellbook Content | |
|  | |
| byte | ID (BF) |
| Short | Size |
| Short | Command (1B) |
| Short | 1 |
| Integer | Serial |
| Short | Graphic |
| Short  byte[8] | Offset  Content |
| **BF.1C – Cast Spell/Last Spell** | |
| Cast Spell/Last Spell  Expansions Flag:  0x0 = LBR&AOS: Mage, Necromancer and Paladin spells;  0x1 = SE: Samurai and Ninja Spells;  0x2 = ML: Spellweaving Spells;  Spell ID: 0x1 – 0x40 – Mage Spells, 0x65 – 0x75 – Necromancer Spells, 0xC9 – 0xD2 – Paladin Spells, 0x91 – 0x96 – Samurai Spells, 0xF5 – 0xFC – Ninja Spells, 0x59 – 0x68 – Spellweaving Spells | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (1C) |
| Short  Integer  byte  byte | Has Spellbook or Spell(2=no spell,1=has spellbook,0=no spellbook, but has spell)  Serial ( if Has Spellbook )  Expansions Flag  Spell ID (if Spell ID = 0, this means last spell) |
| **BF.1D – Design House** | |
| Design House | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (1D) |
| Integer  Integer | House Serial  Revision |
| **BF.1E – Query Design Details** | |
| Query Design Details | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (1E) |
| Integer | House Foundation Serial |
| **BF.2 - Add Walk Key** | |
| Add a key to the top of the Walk Stack | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2) |
| Integer | Key |
|  |  |
| **BF.20 – House Customization** | |
| House Customization is BF Command 20 | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (20) |
| byte | Party Command |
| **BF.20.4 – Begin House Customization** | |
| Begin House Customization | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short  Integer  byte  Short  Short  Short  byte | Command (20)  House Serial  Subcommand(4)  0  0xFFFF  0xFFFF  0xFF |
| **BF.20.5 – End House Customization** | |
| End House Customization | |
|  | |
| Byte | ID (BF) |
| Short | Packet Size |
| Short  Integer  byte  Short  Short  Short  byte | Command (20)  House Serial  Subcommand(5)  0  0xFFFF  0xFFFF  0xFF |
| **BF.21 – Clear Weapon Ability** | |
| Clear Weapon Ability | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (21) |
|  |  |
|  |  |
| **BF.22 – Damage Packet – Old Client** | |
| Damage Packet | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (22) |
| byte  Integer | 1  Serial |
| byte | Amount |
| **BF.24 – Unknown** | |
| Unknown packet, possible it works as OSI detector of ‘third-party’ programs, like sniffers, assistants and etc. | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short  byte | Command (24)  unknown |
| **BF.25 – Enable/Disable SE Spell Icons** | |
| Enable/Disable SE Spell Icons | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (25) |
| byte  byte  byte | 1  Spell ID  Enable/Disable |
| **BF.26 – Set speed mode for movement** | |
| Set speed mode for movement  Speed Mode: 0x0 = Normal movement, 0x1 = Fast movement, 0x2 = Slow movement, 0x3 and above = Hybrid movement | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (26) |
| byte | Speed Mode |
| **BF.2A – Change Race Request** | |
| Change Race Server packet. Note: Currently works only in 2D client | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2A) |
| byte  byte | Female (1 = true, 0 = false)  Race (1 = human, 2 = elf, 255 = error) |
| **BF.2A – Change Race Response** | |
| Change Race Client packet | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2A) |
| Short  Short  Short  Short  Short | Skin Color  Hair Style  Hair Color  Beard Style  Beard ID |
| **BF.2B – Set Mobile Animation** | |
| Change Mobile Animation Pose, server side packet | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2B) |
| Short  byte  byte | Serial (last two bytes of serial, may be OSI mistake)  Animation ID  Frame Count |
| **BF.2C – Use Targeted Item** | |
| Use Targeted Item, client side packet | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2C) |
| Integer  Integer | Item Serial  Target Serial |
| **BF.2D – Cast Targeted Spell** | |
| Cast Targeted Spell client side packet  Spell ID: 0x1 – 0x40 – Mage Spells, 0x65 – 0x75 – Necromancer Spells, 0xC9 – 0xD2 – Paladin Spells, 0x91 – 0x96 – Samurai Spells, 0xF5 – 0xFC – Ninja Spells, 0x59 – 0x68 – Spellweaving Spells | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2D) |
| Short  Integer | Spell ID  Target Serial |
| **BF.2E – Use Targeted Skill** | |
| Use Targeted Skill client side packet | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2E) |
| Short  Integer | Skill ID (ID: from 1 to 55 at present time, if Skill ID = 0, this means that it’s last skill)  Target Serial |
| **BF.2F – KR House Menu Gump Response** | |
| KR House Menu Gump Response | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short  Integer  Integer  Short | Command (2F)  Mobile Serial  House Serial  Subcommand |
| **BF.2F.63 – KR Default House Menu Gump Response** | |
| Default response, also sends in initial house gump | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (63)  0 |
| **BF.2F.65 – KR Change Public/Private House Menu Gump Response** | |
| Change public / private response, sends on toggle public/private button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (65)  0 |
| **BF.2F.66 – KR Convert into the customizable House Menu Gump Response** | |
| Convert into the customizable response, sends on press ‘convert into the customizable’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (66)  0 (1 for server-side packet) |
| **BF.2F.68 – KR Relocate Moving Crate House Menu Gump Response** | |
| Relocate moving crate response, sends on press ‘relocate moving crate’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (68)  0 (1 for server-side packet) |
| **BF.2F.69 – KR Change Sign House Menu Gump Response** | |
| Change sign response, sends on press on sign images. Note: server-side is sending with default response (63) | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (69)  Sign Item ID |
| **BF.2F.6A – KR Change Sign Hanger House Menu Gump Response** | |
| Change sign hanger response, sends on press on sign hanger images. Note: server-side is sending with default response (63) | |
|  | |
| Byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (6A)  Sign Hanger Item ID |
| **BF.2F.6B – KR Change Sign Post House Menu Gump Response** | |
| Change sign post response, sends on press on sign post images. Note: server-side is sending with default response (63) | |
|  | |
| Byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (6B)  Sign Post Item ID |
| **BF.2F.6C – KR Change Foundation Style House Menu Gump Response** | |
| Change foundation style response, sends on press on foundation images. Note: server-side is sending with default response (63) | |
|  | |
| Byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (6C)  Foundation Item ID |
| **BF.2F.6D – KR Rename House Menu Gump Response** | |
| Rename house response, sends on press ‘rename house’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (6D)  0 (1 for server-side packet) |
| **BF.2F.6E – KR Demolish House Menu Gump Response** | |
| Demolish house response, sends on press ‘demolish house’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (6E)  0 (1 for server-side packet) |
| **BF.2F.6F – KR Trade House Menu Gump Response** | |
| Trade house response, sends on press ‘trade house’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (6F)  0 (1 for server-side packet) |
| **BF.2F.70 – KR Make Primary House Menu Gump Response** | |
| Make primary house response, sends on press ‘make primary’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (70)  0 (1 for server-side packet) |
| **BF.2F.71 – KR Change To Co-Owner House Menu Gump Response** | |
| Change player to co-owner response, sends on press ‘change to co-owner’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (71)  Player Serial |
| **BF.2F.72 – KR Change To Friend House Menu Gump Response** | |
| Change player to friend response, sends on press ‘change to friend’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (72)  Player Serial |
| **BF.2F.73 – KR Change To Access House Menu Gump Response** | |
| Change player to access response, sends on press ‘change to access’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (73)  Player Serial |
| **BF.2F.74 – KR Ban House Menu Gump Response** | |
| Ban player response, sends on press ‘ban’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (74)  0 |
| **BF.2F.75 – KR Remove Co-Owner House Menu Gump Response** | |
| Remove player from co-owners list response, sends on press ‘remove co-owner’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (75)  Player Serial |
| **BF.2F.76 – KR Remove Friend House Menu Gump Response** | |
| Remove player from friends list response, sends on press ‘remove friend’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (76)  Player Serial |
| **BF.2F.77 – KR Remove Access House Menu Gump Response** | |
| Remove player from access list response, sends on press ‘remove access’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (77)  Player Serial |
| **BF.2F.78 – KR Remove Ban House Menu Gump Response** | |
| Remove player from ban list response, sends on press ‘remove ban’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (78)  Player Serial |
| **BF.2F.79 – KR Clear Co-Owners List House Menu Gump Response** | |
| Clear co-owners list response, sends on press ‘clear co-owners list’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (79)  0 (1 for server-side packet) |
| **BF.2F.7A – KR Clear Friends List House Menu Gump Response** | |
| Clear friends list response, sends on press ‘clear friends list’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (7A)  0 (1 for server-side packet) |
| **BF.2F.7B – KR Clear Access List House Menu Gump Response** | |
| Clear access list response, sends on press ‘clear access list’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (7B)  0 (1 for server-side packet) |
| **BF.2F.7C – KR Clear Bans List House Menu Gump Response** | |
| Clear bans list response, sends on press ‘clear bans list’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (7C)  0 (1 for server-side packet) |
| **BF.2F.7D – KR Add Access House Menu Gump Response** | |
| Add access response, sends on press ‘add access’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (7D)  0 |
| **BF.2F.7E – KR Valid Add Access House Menu Gump Response** | |
| Valid add access response, sends on valid targeting for ‘add access’. Only server-side packet. | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (7E)  Player Serial |
| **BF.2F.7F – KR Invalid Add Access House Menu Gump Response** | |
| Invalid add access response, sends on invalid targeting for ‘add access’. Only server-side packet. | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (7F)  0 |
| **BF.2F.80 – KR Customize House Menu Gump Response** | |
| Customize house response, sends on press ‘customize house’ button | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (2F) |
| Integer  Integer  Short  Integer | Mobile Serial  House Serial  Subcommand (80)  House Foundation Serial (0 for client-side packet) |
| **BF.30 – KR Target By Resource Macro** | |
| KR Macro ‘Target By Resource’ since 2.46.0.3  Resource Type values: 0 – ore, 1 – sand, 2- wood, 3 – graves, 4 – red mushrooms | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (30) |
| Integer  Short | Tool Serial  Resource Type |
| **BF.31 – KR Start Hotbar Slot Timer** | |
| KR Start Hotbar Slot Timer since 2.48.0.3  Start timer for hotbar slot with object having ItemID from packet  Note 1: Timer will work for all slot with objects having ItemID from packet  Note 2: Currently (2.48.0.7) only bandages itemid work. ItemIDs are 0xE21 and 0xEE9 | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short  Short | Command (31)  0x1 |
| Integer  Integer | ItemID  Duration |
| **BF.4 - Close Gump** | |
| Close gump | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (4) |
| Integer  Integer | Type ID  Button ID |
|  |  |
| **BF.5 – Screen Size** | |
| Screen Size | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (5) |
| Integer  Integer | Width  Height |
| **BF.6 – Party** | |
| Party system is BF Command 6 | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command |
|  |  |
| **BF.6.1 – Add Member** | |
| Add member to party | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command (1) |
| Integer | Serial |
|  |  |
| **BF.6.1 – Displays members list** | |
| Displays members list of the party | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command (1) |
| byte  loop | Number of Members  Member |
| Integer  endloop | Serial  Member |
| **BF.6.2 - Remove Member** | |
| Remove a member from your party | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command (2) |
| byte | Number of Members |
| Integer  loop  Integer  endloop | Serial of removed member  Member  Serial  Member |
| **BF.6.3 - Party Private Msg** | |
| Tell a party member a message | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command (3) |
| Integer | Serial |
| Short[\*] | Unicode Message |
|  |  |
| **BF.6.4 - Party Chat** | |
| Send a message to entire party. (Serial not present in Client Msg) | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command (4) |
| Integer | Serial of source |
| Short[\*] | Unicode Text |
|  |  |
| **BF.6.6 - Party Loot** | |
| Party can loot me | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command (6) |
| byte | Can Loot |
|  |  |
| **BF.6.7 - Party Invitation** | |
| Party Invitation | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command (7) |
| Integer | Party Leader Serial |
| **BF.6.8 - Accept** | |
| Accept a join invitation | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command (8) |
| Integer | Party Leader Serial |
|  |  |
| **BF.6.9 - Decline** | |
| Decline join invitation | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (6) |
| byte | Party Command (9) |
| Integer | Party Leader Serial |
|  |  |
| **BF.7 – Quest Arrow** | |
| Quest Arrow | |
|  | |
| byte | BF |
| Short | Packet Size |
| Short | Command (7) |
| byte | Right Click (1=yes, 0=no) |
| **BF.8 – Map Change** | |
| Map Change | |
|  | |
| byte | BF |
| Short | Packet Size |
| Short | Command (8) |
| byte | Map ID (0=felucca, 1= trammel, 2=ilshenar, 3=malas, 4=tokuno) |
|  |  |
| **BF.9 – Disarm Request** | |
| Disarm Request | |
|  | |
| byte | BF |
| Short | Packet Size |
| Short | Command (9) |
| byte  Integer  Short | 2  6  0 |
| **BF.A – Stun Request** | |
| Stun Request | |
|  | |
| byte | BF |
| Short | Packet Size |
| Short | Command (A) |
| Byte[\*] | Unknown |
| **BF.B - Client Language** | |
| Set client language | |
|  | |
| byte | ID (BF) |
| Short | Packet Size |
| Short | Command (B) |
| byte[4] | Language |
|  |  |
| **BF.C - Close Status** | |
| Close status gump | |
|  | |
| Byte | ID (BF) |
| Short | Packet Size |
| Short | Command (C) |
| Integer | Serial |
|  |  |
| **BF.E - Animate** | |
| Animate | |
|  | |
| Byte | ID (BF) |
| Short | Packet Size |
| Short | Command (E) |
| Integer | Action |
| **BF.F - Empty** | |
| Empty. Real purpose is unknown. | |
|  | |
| Byte | ID (BF) |
| Short | Packet Size |
| Short | Command (F) |
|  |  |
| **C0 - Effect Hued FX** | |
| Hued effect 0x24 bytes  Type:  00 = Go from source to destination  01 = Lightning strike  02 = Stay at location  03 = Stay with source | |
|  | |
| byte | ID (C0) |
| byte  Integer  Integer  Short  Short  Short  char | Type  Serial Source  Serial Destination  Item ID  X Source  Y Source  Z Source |
| Short  Short  char  byte  byte  Short  byte  byte  Integer  Integer | X Destination  Y Destination  Z Destination  Speed  Duration  0  Fixed Duration  Explode  Hue  Render Mode |
| **C1 – Localized Message** | |
| Display a cliloc message  Type: 0=say,1=system,2=emote,6=label,7=focus, 8=whipser,9=yell,10=spell,13=guild,14=alliance,15=GM,0xc0=encoded commands | |
|  | |
| byte | ID (C1) |
| Short | Packet Size |
| Integer | Serial |
| Short | Body |
| byte | Type |
| Short | Hue |
| Short | Font |
| Integer | Message Number |
| char[30] | Name |
| char[\*] | Arguments |
|  |  |
| **C2 - Text Entry Unicode** | |
| Unicode text entry | |
|  | |
| byte | ID (C2) |
| byte[2] | Packet Size |
| Integer  Integer  Integer  Integer  Short | Serial1  Serial2  0  0  0 |
|  |  |
| **C3 - GQ Request** | |
| GQ Request | |
|  | |
| byte | ID (C3) |
| byte[2] | Packet Size |
| Integer  Integer  Integer  Integer | 1  ID  Customer ID  Serial |
| Integer  Short  char[length]  Integer  Integer  Integer  Integer  Integer  Integer  Integer  Integer  Integer  byte  byte  char[3]  byte[\*] | 0  length  Server Name  Call time (in seconds)  Map (0=felucca,1=trammel,2=ilshenar,3=malas,4=tokuno)  X  Y  Z  Volume  Rank  -1  Type  1  1  Language  Text |
| **C4 - Semi Visible** | |
| Semivisible 6 bytes | |
|  | |
| byte | ID (C4) |
| byte[5] | Unknown |
|  |  |
| **C5 - Invalid Map** | |
| Invalid map 0xCB bytes | |
|  | |
| byte | ID (C5) |
| byte[0xCA] | Unknown |
|  |  |
| **C6 - Invalid Map Enable** | |
| Enable invalid map 1 byte | |
|  | |
| byte | ID (C6) |
|  |  |
| **C7 - Particle Effect** | |
| Particle Effect 0x31 bytes Effect ID and Explode Effect ID correspond directly to the pbg files in the Particles directory. | |
|  | |
| byte | ID (C7) |
| byte | Type (00 = moving effect, 01 = lightning effect, 02 = item effect, 03 = static effect) |
| Integer | Serial Source |
| Integer | Serial Destination |
| Short | Tile ID |
| Short | X Source |
| Short | Y Source |
| byte | Z Source |
| Short | X Destination |
| Short | Y Destination |
| byte | Z Destination |
| byte | Speed |
| byte | Duration |
| byte | 0 |
| byte | 0 |
| byte | Fixed Duration |
| byte | Explode |
| Integer | Hue |
| Integer | Render Mode |
| Short | Effect ID |
| Short | Explode Effect ID |
| Short | Explode Sound |
| Integer | Serial |
| byte | Layer |
| Short | unknown |
| **C8 - Update Range Change** | |
| Update range change 2 bytes  This packet says how far away the client wants to see.. that allows slower modems to not get items and MOBs that are over a certain distance away. | |
|  | |
| byte | ID (C8) |
| byte | Distance |
|  |  |
| **C9 - Trip Time** | |
| Trip time 6 bytes | |
|  | |
| byte | ID (C9) |
| byte | Unknown |
| Integer | Unknown |
| **CA - UTrip time** | |
| Utrip time 6 bytes | |
|  | |
| byte | ID (CA) |
| byte | Unknown |
| Integer | Unknown |
| **CB - GQ Count** | |
| GQ Count 7 bytes | |
|  | |
| byte | ID (CB) |
| Short | Unknown |
| Integer | Count |
| **CC – Localized Message Plus String** | |
| Cliloc Message and String  Type: 0=say,1=system,2=emote,6=label,7=focus, 8=whipser,9=yell,10=spell,13=guild,14=alliance,15=GM,0xc0=encoded commands  Affix Type: 0x0 = Append, 0x1 = Prepend, 0x2 = System | |
|  | |
| byte | ID (CC) |
| Short | Packet Size |
| Integer  Short | Serial  Graphic |
| byte  Short  Short  Integer  byte  char[30]  char[\*]  char[\*] | Type  Hue  Font  Number  Affix Type  Name  Affix  Arguments |
| **CD - Unknown** | |
| Unknown Packet | |
|  | |
| byte | ID (CD) |
| byte[\*] | Unknown |
| **CE - Unknown** | |
| Unknown Packet | |
|  | |
| byte | ID (CE) |
| byte[\*] | Unknown |
| **CF – Account Login 2** | |
| Login to the login server. At present, OSI really uses 0x80 packets, but this one absolutely identified to 0x80. 0x3E bytes | |
|  | |
| byte | ID (CF) |
| char[30]  char[30]  byte | Name  Password  Unknown |
| **D0 – Configuration File** | |
| Send to server configuration file  2 bytes | |
|  | |
| byte | ID (D0) |
| byte[\*] | Unknown |
|  |  |
| **D1 - Logout Status** | |
| Logout Status (Server version)  Client version not send anything except packet ID (D1): it’s Logout Request. Answer to this request is this server-side packet.  2 bytes | |
|  | |
| byte | ID (D1) |
| byte | 1 |
|  |  |
| **D2 - Unknown** | |
| Unknown packet | |
|  | |
| byte | ID (D2) |
| byte[\*] | Unknown |
| **D3 - Unknown** | |
| Unknown packet | |
|  | |
| byte | ID (D3) |
| byte[\*] | Unknown |
| **D4 – Book Header** | |
| Book Header | |
|  | |
| byte | ID (D4) |
| Short  Integer  byte  byte  Short  Short  char[\*]  byte  Short  char[\*]  byte | Packet Size  Serial  1  Writeable  Pages Count  Title Length + 1  Title  0  Author Length + 1  Author  0 |
| **D5 - Unknown** | |
| Unknown packet | |
|  | |
| byte | ID (D5) |
| byte[\*] | Unknown |
| **D6 – Batch Query Properties** | |
| Batch Query Properties (Client version)  Client sends serials of items from 0x3C packet and server sends Object Properties packet as answer | |
|  | |
| byte | ID (D6) |
| Short  loop  Integer  endloop | Packet Size  Item Info; count = (Packet Size-3)/4  Serial  Item Info |
| **D6 – Object Properties** | |
| Object Properties (Server version) | |
|  | |
| byte | ID (D6) |
| Short  Short  Integer  Short  loop  Integer  Short  byte[\*]  endloop  Integer | Packet Size  1  Serial  0  Property  Number  Arguments Length  Arguments  Property  Hash |
| **D7 - Generic Command** | |
| Miscellaneous Commands | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command |
|  |  |
| **D7.10 – Designer Clear** | |
| Designer Clear | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (10) |
| byte | 0xA |
| **D7.12 – Designer Level** | |
| Designer Level | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short  byte  Integer | Command (12)  0  Level |
| byte | 0xA |
| **D7.13 – Designer Roof** | |
| Designer Roof | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short  byte  Integer  byte  Integer  byte  Integer  byte  Integer | Command (13)  0  Item ID  0  X  0  Y  0  Z |
| byte | 0xA |
| **D7.14 – Designer Roof Delete** | |
| Designer Roof Delete | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short  byte  Integer  byte  Integer  byte  Integer  byte  Integer | Command (14)  0  Item ID  0  X  0  Y  0  Z |
| byte | 0xA |
| **D7.19 – Set Weapon Ability** | |
| Set Weapon Ability | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short  byte | Command (19)  0 |
| Integer | Ability Index |
| byte | 0xA |
| **D7.1A – Designer Revert** | |
| Designer Revert | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (1A) |
| byte | 0xA |
| **D7.1E – Equip Last Weapon** | |
| Equip Last Weapon | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (1E) |
| byte | 0xA |
| **D7.2 – Designer Backup** | |
| Designer Backup | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (2) |
| byte | 0xA |
| **D7.3 – Designer Restore** | |
| Designer Restore | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (3) |
| byte | 0xA |
| **D7.4 – Designer Commit** | |
| Designer Commit | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (4) |
| byte | 0xA |
| **D7.5 – Designer Delete** | |
| Designer Delete | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short  byte  Integer  byte  Integer  byte  Integer  byte  Integer  byte | Command (5)  0  Item ID  0  X  0  Y  0  Z  0xA |
|  |  |
| **D7.6 – Designer Build** | |
| Designer Build | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short  byte  Integer  byte  Integer  byte  Integer  byte | Command (6)  0  Item ID  0  X  0  Y  0xA |
|  |  |
| **D7.28 – Guild Button Request** | |
| Guild Button Request | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (28) |
| byte | 0xA |
| **D7.32 – Quests Button Request** | |
| Quests Button Request | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (32) |
| byte | 0xA |
| **D7.C – Designer Close** | |
| Designer Close | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (C) |
| byte | 0xA |
| **D7.D – Designer Stairs** | |
| Designer Stairs | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short  byte  Integer  byte  Integer  byte  Integer | Command (D)  0  Stair ID  0  X  0  Y |
| byte | 0xA |
| **D7.E – Designer Synch** | |
| Designer Synch | |
|  | |
| byte | ID (D7) |
| Short  Integer | Packet Size  Serial |
| Short | Command (E) |
| byte | 0xA |
| **D8 – Design State Detailed** | |
| Design State Detailed  Server sends to client design data: tiles and stairs in compression mode. OSI uses ZLib.dll with Z\_DEFAULT\_COMPRESSION compression level.  Flags = (((Size >> 4) & 0xF0) | ((Length >> 8) & 0xF)) ) | |
|  | |
| byte | ID (D8) |
| Short  byte | Packet Size  Compression Type (at present time, it’s 0x03) |
| byte  Integer  Integer | 0  Serial  Revision |
| Short  Short  byte  loop  byte  byte  byte  byte  byte[\*]  endloop  loop  byte  byte  byte  byte  byte[\*]  endloop | Tiles Length  Buffer Length  Plane Count  Plane  (0x20 | Plane Index)  Plane Size  Plane Length  Flags  Plane Buffer  Plane  Stairs  (9+Plane Index)  Stairs Size  Stairs Length  Flags  Stairs Buffer  Stairs |
| **D9 – Hardware Info** | |
| Hardware Info  Client Type: 1 = client version < 4.0.1a; 0 = client version >= 4.0.1a | |
|  | |
| byte | ID (D9) |
| Short  byte  Integer  Integer  Integer  Integer  byte  Integer  Integer  Integer  byte  Integer  Integer  Integer  Integer  Short  Short  char[64]  Integer  Integer  Integer  byte  byte  byte  byte  char[4]  char[64] | Packet Size  Client Type  Instance ID  OS Major  OS Minor  OS Revision  CPU Manufacturer  CPU Family  CPU Model  CPU Clock Speed  CPU Quantity  Physical Memory  Screen Width  Screen Height  Screen Depth  DirectX Major  DirectX Minor  Video Card Description  Video Card Vendor ID  Video Card Device ID  Video Card Memory  Distribution  Clients Running  Clients Installed  Partial Installed  Language  Unknown |
| **DA – Mahjong Game Commands** | |
| Mahjong Game Commands. Note: Currently works only in 2D client. | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short | Command |
| **DA.10 – Mahjong Game Open Seat** | |
| Mahjong Game Open Seat | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short  byte | Command(10)  Seat Position |
| **DA.11 – Mahjong Game Change Options** | |
| Mahjong Game Change Options  Options: 0x1 = Show Scores, 0x2 = Spectator Vision | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short  Integer | Command(11)  Options |
| **DA.15 – Mahjong Game Move Wall Break Indicator** | |
| Mahjong Game Move Wall Break Indicator | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short  Short  Short | Command(15)  Y  X |
| **DA.16 – Mahjong Game Toggle Public Hand** | |
| Mahjong Game Toggle Public Hand | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short  Integer | Command(16)  Public Hand (1=yes, 0=no) |
| **DA.17 – Mahjong Game Move Tile** | |
| Mahjong Game Move Tile | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short  byte  byte  byte  byte  byte  Short  Short  byte  Short  Short  byte | Command(17)  Number  Current Direction  New Direction  0  Flip  Current Y  Current X  1  New Y  New X  0 |
| **DA.18 – Mahjong Game Move Dealer Indicator** | |
| Mahjong Game Move Dealer Indicator | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short  byte  byte  Short  Short | Command(18)  Direction  Wind  Y  X |
| **DA.19 – Mahjong Game Join Game** | |
| Mahjong Game Join Game | |
|  | |
| byte | ID (DA) |
| Short  Integer  Short | Packet Size  Game Serial  Command(19) |
| **DA.1A – Mahjong Game Relieve** | |
| Mahjong Game Relieve | |
|  | |
| byte | ID (DA) |
| Short  Integer  Short | Packet Size  Game Serial  Command(1A) |
| **DA.2 – Mahjong Game Players Info** | |
| Mahjong Game Players Info | |
|  | |
| byte | ID (DA) |
| Short  Integer  Short  Short  loop  Integer  byte  byte  Integer  Short  Short  char[30]  byte  endloop | Packet Size  Game Serial  Command(2)  Player Seats Count  Players  Player Serial  Dealer Position (1=yes,2=no)  Player Index  Player Score  0  Public (0x1=yes,0x0=no)  Player Name  Player In Game(1=no,0=yes)  Players |
| **DA.3 – Mahjong Game Tile Info** | |
| Mahjong Game Tile Info | |
|  | |
| byte | ID (DA) |
| Short  Integer  Short  byte  byte  Short  Short  byte  byte  byte | Packet Size  Game Serial  Command(3)  Tile Number  Tile Value  Y  X  Stack Level  Direction  Flipped (0x10=yes,0x0=no) |
| **DA.4 – Mahjong Game Tiles Info** | |
| Mahjong Game Tiles Info | |
|  | |
| byte | ID (DA) |
| Short  Integer  Short  byte  loop  byte  byte  Short  Short  byte  byte  byte  endloop | Packet Size  Game Serial  Command(4)  Tiles Length  Tiles  Tile Number  Tile Value  Y  X  Stack Level  Direction  Flipped (0x10=yes,0x0=no)  Tiles |
| **DA.5 – Mahjong Game General Info** | |
| Mahjong Game General Info  Options = Show Scores (0x1=yes,0x0=no) | Spectator Vision (0x2=yes,0x0=no) | |
|  | |
| byte | ID (DA) |
| Short  Integer  Short  Short  Short  byte  byte  byte  Short  Short  byte  Short  Short | Packet Size  Game Serial  Command(5)  0  Options  First Dices  Second Dices  Dealer Indicator Wind  Dialer Indicator Y  Dialer Indicator X  Dialer Indicator Direction  Wall Break Indicator Y  Wall Break Indicator X |
| **DA.6 – Mahjong Game Leave Game** | |
| Mahjong Game Leave Game | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short | Command(6) |
| **DA.A – Mahjong Game Give Points** | |
| Mahjong Game Give Points | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short  byte  Integer | Command(A)  Position  Points |
| **DA.B – Mahjong Game Roll Dices** | |
| Mahjong Game Roll Dices | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short | Command(B) |
| **DA.C – Mahjong Game Build Walls** | |
| Mahjong Game Build Walls | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short | Command(C) |
| **DA.D – Mahjong Game Reset Scores** | |
| Mahjong Game Reset Scores | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short | Command(D) |
| **DA.F – Mahjong Game Assign Dealer** | |
| Mahjong Game Assign Dealer | |
|  | |
| byte | ID (DA) |
| Short  Integer | Packet Size  Game Serial |
| Short  byte | Command(F)  Position |
| **DB - Unknown** | |
| Unknown packet | |
|  | |
| byte | ID (DB) |
| byte[\*] | Unknown |
| **DC – OPLInfo – since 4.0.5a client** | |
| OPLInfo packet. | |
|  | |
| byte | ID (DC) |
| Integer  Integer | Serial  Hash |
| **DD – Generic Gump – since 5.0.0a client** | |
| Generic Gump packet.  OSI uses this one since launching of ML expansion. They use ZLib.dll with Z\_BEST\_SPEED compression level. | |
|  | |
| byte | ID (DD) |
| Short  Integer  Integer  Integer  Integer  Integer  Integer  byte[\*]  Integer  Integer  Integer  byte[\*] | Packet Size  Gump Serial  Gump Type ID  Gump X  Gump Y  Gump Entries Length + 4  Layout Length  Compressed Gump Entries  Lines Count  Gump Strings Length + 4  Uncompressed Gump Strings Length  Compressed Gump Strings |
| **DE – Update Mobile Status** | |
| Update Mobile Status packet  OSI uses this one since launching of buff/debuff system. | |
|  | |
| byte | ID (DE) |
| Short  Integer  byte | Packet Size  Mobile Serial  Status (usually 0, i got 1 only after attack (from) another character |
| **DF – Update Attribute** | |
| Update Attribute packet  Attributes: BonusStr = 0x01, BonusDex = 0x02, BonusInt = 0x03, BonusHits = 0x07, BonusStamina = 0x08, BonusMana = 0x09, RegenHits = 0x0A, RegenStam = 0x0B, RegenMana = 0x0C, NightSight = 0x0D, Luck = 0x0E, ReflectPhysical = 0x10, EnhancePotions = 0x11, AttackChance = 0x12, DefendChance = 0x13, SpellDamage = 0x14, CastRecovery = 0x15, CastSpeed = 0x16, ManaCost = 0x17, ReagentCost = 0x18, WeaponSpeed = 0x19, WeaponDamage = 0x1A, PhysicalResistance = 0x1B, FireResistance = 0x1C, ColdResistance = 0x1D, PoisonResistance = 0x1E, EnergyResistance = 0x1F, MaxPhysicalResistance = 0x20, MaxFireResistance = 0x21, MaxColdResistance = 0x22, MaxPoisonResistance = 0x23, MaxEnergyResistance = 0x24, AmmoCost = 0x26, KarmaLoss = 0x28,  OSI uses this one since 5.0.2b client. | |
|  | |
| byte | ID (DF) |
| Short  Integer  Short  Short  loop  Short  Integer  Short  byte[9]  Integer  byte[14]  endloop | Packet Size  Player Serial  Attribute ID  Items Count  Items  Base Value  0  Delta Value  0  Item Label Number  0  Items |
| **DF – Buff/Debuff** | |
| Buff/Debuff packet  OSI uses this one since 5.0.2b client. | |
|  | |
| byte | ID (DF) |
| Short  Integer  Short  Short  Integer  Short  Short  Integer  Short  byte[3]  Integer  Integer  Integer  Short  Short  Short  char[length\*2]  Short  Short | Packet Size  Player Serial  Buff Icon ID  Type  If ( Type = 0x01 )  0  Buff Icon ID  1  0  Buff Duration in seconds  0  Buff Title Cliloc  Buff Secondary Cliloc  0  Arguments Mode  0  If (Arguments Mode = 0x01)  length  Arguments  1  EndIf Arguments Mode  0  EndIf Type |
| **E0 – Bug Report** | |
| Bug Report packet, client side  Bug Category: 0x01 - World Environment; 0x02 – Wearables; 0x03 – Combat; 0x04 – UI; 0x05 – Crash; 0x06 – Stuck; 0x07 – Animations; 0x08 – Performance; 0x09 – NPCs; 0x0A – Creatures; 0x0B – Pets; 0x0C – Housing; 0x0D - Lost Item; 0x0E – Exploit; 0x0F – Other  Only KR packet | |
|  | |
| byte | ID (E0) |
| Short  byte[3]  byte  Short  char[\*] | Packet Size  Language  0  Bug Category  Bug Description in Unicode |
| **E1 – KR Character List Update** | |
| KR Character List Update, client side, only in character list menu  Only KR packet  Notes: Probably KR Client Flags (always 0x2) is offset for 0x8D packet flags – character creation. | |
|  | |
| byte | ID (E1) |
| Short  Short  Integer | Packet Size  1  KR Client Flags |
| **E2 – Mobile Status/Animation Update** | |
| Mobile Status/Animation Update, server side | |
|  | |
| byte | ID (E2) |
| Integer  Short  byte  Short | Mobile Serial  Action  0  Count |
| **E3 – KR Encryption Request** | |
| KR Encryption Request, server side  Notes: Packet Size must be 77 bytes. A[0] is always 0x2. B[0] = 0x2, B[1] = 0x11, B[2] = 0x0.  A, B, C, D, E – parameters for KR game encryption – AES in CFB Mode  Only KR packet | |
|  | |
| Byte | ID (E3) |
| Integer  byte[lengthA]  Integer  byte[length B]  Integer  byte[length C]  Integer  Integer  byte[length E] | length A  A  length B  B  length C  C  D  length E  E |
| **E4 – KR Encryption Response** | |
| KR Encryption Response, client side  Notes: Reply to E3 packet from server  A – parameter for KR game encryption – AES in CFB Mode  Only KR packet | |
|  | |
| Byte | ID (E4) |
| Short  Integer  byte[lengthA] | Packet Size  length A  A |
| **E5 – Display Waypoint** | |
| Display Waypoint, server side  Notes: Server sends this packet both to 2D and KR, but only KR will display waypoint on the radar map.  If Ignore Object Serial is true, client will use coordinates from packet and will ignore serial object coordinates. | |
|  | |
| byte | ID (E5) |
| Short  Integer  Short  Short  sbyte  byte  Short  Short  Integer  char[\*]  Short | Packet Size  Object Serial  Object X  Object Y  Object Z  Object Map ID  Object Type  Ignore Object Serial (1 = true, 0 = false)  Object Cliloc Description  Object Cliloc Description Arguments in Unicode  0 |
| **E6 – Hide Waypoint** | |
| Hide Waypoint, server side  Notes: Server sends this packet both to 2D and KR, but only KR will hide waypoint from the radar map | |
|  | |
| byte | ID (E6) |
| Integer | Object Serial |
| **E7 – Continue Highlight KR UI Element** | |
| Continue Highlight KR UI Element, server side  Only KR packet | |
|  | |
| byte | ID (E7) |
| Integer  Short  Integer  byte | Mobile Serial  UI Element ID  Destination Object Serial  1 |
| **E8 – Remove Highlight KR UI Element** | |
| Remove Highlight KR UI Element, client side  Only KR packet. Client sends this packet only if server sent E7 packet before. | |
|  | |
| byte | ID (E8) |
| Integer  Short  Integer  byte  byte | Mobile Serial  UI Element ID  Destination Object Serial  1  1 |
| **E9 – Toggle Highlight KR UI Element** | |
| Toggle Highlight KR UI Element, server side  Description types: ”ToggleInventory”, ”TogglePaperdoll”, ”ToggleMap”, ””  Only KR packet | |
|  | |
| byte | ID (E9) |
| Integer  Short  char[64]  Integer | Mobile Serial  UI Element ID  Description  Command ID |
| **EA – Enable KR Hotbar** | |
| Enable KR Hotbar, server side  Only KR packet | |
|  | |
| byte | ID (EA) |
| Short | Enable (1 = true, 0 = false) |
| **EB – Report Use KR Hotbar Icon** | |
| Report Use KR Hotbar Icon, client side  Only KR packet. Client sends this packet only if server sent EA packet before  Type: 0x1 – spell, 0x2 – weapon ability, 0x3 – skill, 0x4 – item, 0x5 – scroll  Object ID: serial for item, id for other types. Always in reversed mode.  Note: since KR 2.46.\*.\* Object ID is serial for scroll too. | |
|  | |
| Byte | ID (EB) |
| Short  Short  byte  byte  Integer | 1  6  Type  0  Object ID (reversed!!) |
| **EC – Equip Items KR Macro** | |
| Equip Items KR Macro, client side  Only KR packet | |
|  | |
| Byte | ID (EC) |
| Short  byte  loop  Integer  endloop | Packet Size  Items Count  Items  Item Serial  Items |
| **ED – Unequip Items KR Macro** | |
| Unequip Items KR Macro, client side  Only KR packet | |
|  | |
| Byte | ID (ED) |
| Short  byte  loop  Short  endloop | Packet Size  Layers Count  Layers  Layer ID  Layers |
| **F2 – Unknown** | |
| Unknown server-side packet.  A, B and C – unknown values. A is increment value, so may be it’s global counter. Server sends this packet without client request.  Only KR packet since 2.48.0.3 | |
|  | |
| byte | ID (F2) |
| Integer  Short  Short  Integer  Short  Short  Integer  Short  Short | 0x116  A  B  0x116  A  C  0x116  A  C |