Appendix A

KEYSYM Encoding

For convenience, KEYSYM values are viewed as split into four bytes:

- Byte 1 (for the purposes of this encoding) is the most-significant 5 bits (because of the 29-bit effective values)
- Byte 2 is the next most-significant 8 bits
- Byte 3 is the next most-significant 8 bits
- Byte 4 is the least-significant 8 bits

There are two special KEYSYM values: NoSymbol and VoidSymbol.

Byte 1	Byte 2	Byte 3	Byte 4	Name
0	0	0	0	NoSymbol
0	255	255	255	VoidSymbol

All other standard KEYSYM values have zero values for bytes 1 and 2. Byte 3 indicates a character code set, and byte 4 indicates a particular character within that set.

Byte 3	Byte 4
0	Latin 1
1	Latin 2
2	Latin 3
3	Latin 4
4	Kana
5	Arabic
6	Cyrillic
7	Greek
8	Technical
9	Special
10	Publishing
11	APL
12	Hebrew
255	Keyboard

Each character set contains gaps where codes have been removed that were duplicates with codes in previous character sets (that is, character sets with lesser byte 3 value).

The 94 and 96 character code sets have been moved to occupy the right-hand quadrant (decimal 129 through 256), so the ASCII subset has a unique encoding across byte 4, which corresponds to the ASCII character code. However, this cannot be guaranteed with future registrations and does not apply to all of the Keyboard set.

To the best of our knowledge, the Latin, Kana, Arabic, Cyrillic, Greek, APL, and Hebrew sets are from the appropriate ISO and/or ECMA international standards. There are no Technical, Special, or Publishing international standards, so these sets are based on Digital Equipment Corporation standards.

The ordering between the sets (byte 3) is essentially arbitrary. National and international standards bodies were commencing deliberations regarding international 2-byte and 4-byte character sets at the time these keysyms were developed, but we did not know of any proposed layouts.

The order may be arbitrary, but it is important in dealing with duplicate coding. As far as possible, keysym values (byte 4) follow the character set encoding standards, except for the Greek and Cyrillic keysyms which are based on early draft standards. In the Latin-1 to Latin-4 sets, all duplicate glyphs occupy the same code position. However, duplicates between Greek and Technical do not occupy the same code position. Applications that wish to use the Latin-2, Latin-3, Latin-4, Greek, Cyrillic, or Technical sets may find it convenient to use arrays to transform the keysyms.

There is a difference between European and US usage of the names Pilcrow, Paragraph, and Section, as follows:

US name	European name	code position in Latin-1

Section sign	Paragraph sign	10/07
Paragraph sign	Pilcrow sign	11/06

We have adopted the US names (by accident rather than by design).

The Keyboard set is a miscellaneous collection of commonly occurring keys on keyboards. Within this set, the keypad symbols are generally duplicates of symbols found on keys on the main part of the keyboard, but they are distinguished here because they often have a distinguishable semantics associated with them.

Keyboards tend to be comparatively standard with respect to the alphanumeric keys, but they differ radically on the miscellaneous function keys. Many function keys are left over from early timesharing days or are designed for a specific application. Keyboard layouts from large manufacturers tend to have lots of keys for every conceivable purpose, whereas small workstation manufacturers often add keys that are solely for support of some of their unique functionality. There are two ways of thinking about how to define keysyms for such a world:

- The Engraving approach
- The Common approach

The Engraving approach is to create a keysym for every unique key engraving. This is effectively taking the union of all key engravings on all keyboards. For example, some keyboards label function keys across the top as F1 through Fn, and others label them as PF1 through PFn. These would be different keys under the Engraving approach. Likewise, Lock would differ from Shift Lock, which is different from the up-arrow symbol that has the effect of changing lowercase to uppercase. There are lots of other aliases such as Del, DEL, Delete, Remove, and so forth. The Engraving approach makes it easy to decide if a new entry should be added to the keysym set: if it does not exactly match an existing one, then a new one is created. One estimate is that there would be on the order of 300–500 Keyboard keysyms using this approach, without counting foreign translations and variations.

The Common approach tries to capture all of the keys present on an interesting number of keyboards, folding likely aliases into the same keysym. For example, Del, DEL, and Delete are all merged into a single keysym. Vendors would be expected to augment the keysym set (using the vendor-specific encoding space) to include all of their unique keys that were not included in the standard set. Each vendor decides which of its keys map into the standard keysyms, which presumably can be overridden by a user. It is more difficult to implement this approach, because judgment is required about when a sufficient set of keyboards implements an engraving to justify making it a keysym in the standard set and about which engravings should be merged into a single keysym. Under this scheme there are an estimated 100–150 keysyms.

Although neither scheme is perfect or elegant, the Common approach has been selected because it makes it easier to write a portable application. Having the Delete functionality merged into a single keysym allows an application to implement a deletion function and expect reasonable bindings on a wide set of workstations. Under the Common approach, application writers are still free to look for and interpret vendor-specific keysyms, but because they are in the extended set, the application developer is more conscious that they are writing the application in a nonportable fashion.

In the listings below, Code Pos is a representation of byte 4 of the KEYSYM value, expressed as most-significant/least-significant 4-bit values. The Code Pos numbers are for reference only and do not affect the KEYSYM value. In all cases, the KEYSYM value is:

byte3 * 256 + byte4

Byte 3	Byte 4	Code Pos	Name	Set
000	032	02/00	SPACE	Latin-1
000	033	02/01	EXCLAMATION POINT	Latin-1
000	034	02/02	QUOTATION MARK	Latin-1
000	035	02/03	NUMBER SIGN	Latin-1
000	036	02/04	DOLLAR SIGN	Latin-1
000	037	02/05	PERCENT SIGN	Latin-1
000	038	02/06	AMPERSAND	Latin-1
000	039	02/07	APOSTROPHE	Latin-1
000	040	02/08	LEFT PARENTHESIS	Latin-1
000	041	02/09	RIGHT PARENTHESIS	Latin-1
000	042	02/10	ASTERISK	Latin-1
000	043	02/11	PLUS SIGN	Latin-1
000	044	02/12	COMMA	Latin-1
000	045	02/13	MINUS SIGN	Latin-1
000	046	02/14	FULL STOP	Latin-1
000	047	02/15	SOLIDUS	Latin-1

Byte 3	Byte 4	Code Pos	Name	Set
000	048	03/00	DIGIT ZERO	Latin-1
000	049	03/01	DIGIT ONE	Latin-1
000	050	03/02	DIGIT TWO	Latin-1
000	051	03/03	DIGIT THREE	Latin-1
000	052	03/04	DIGIT FOUR	Latin-1
000	053	03/05	DIGIT FIVE	Latin-1
000	054	03/06	DIGIT SIX	Latin-1
000	055	03/07	DIGIT SEVEN	Latin-1
000	056	03/08	DIGIT EIGHT	Latin-1
000	057	03/09	DIGIT NINE	Latin-1
000	058	03/10	COLON	Latin-1
000	059	03/11	SEMICOLON	Latin-1
000	060	03/12	LESS THAN SIGN	Latin-1
000	061	03/13	EQUALS SIGN	Latin-1
000	062	03/14	GREATER THAN SIGN	Latin-1
000	063	03/15	QUESTION MARK	Latin-1
000	064	04/00	COMMERCIAL AT	Latin-1
000	065	04/01	LATIN CAPITAL LETTER A	Latin-1
000	066	04/02	LATIN CAPITAL LETTER B	Latin-1
000	067	04/03	LATIN CAPITAL LETTER C	Latin-1
000	068	04/04	LATIN CAPITAL LETTER D	Latin-1
000	069	04/05	LATIN CAPITAL LETTER E	Latin-1
000	070	04/06	LATIN CAPITAL LETTER F	Latin-1
000	071	04/07	LATIN CAPITAL LETTER G	Latin-1
000	072	04/08	LATIN CAPITAL LETTER H	Latin-1
000	073	04/09	LATIN CAPITAL LETTER I	Latin-1
000	074	04/10	LATIN CAPITAL LETTER J	Latin-1
000	075	04/11	LATIN CAPITAL LETTER K	Latin-1
000	076	04/12	LATIN CAPITAL LETTER L	Latin-1
000	077	04/13	LATIN CAPITAL LETTER M	Latin-1
000	078	04/14	LATIN CAPITAL LETTER N	Latin-1
000	079	04/15	LATIN CAPITAL LETTER O	Latin-1
000	080	05/00	LATIN CAPITAL LETTER P	Latin-1
000	081	05/01	LATIN CAPITAL LETTER Q	Latin-1
000	082	05/02	LATIN CAPITAL LETTER R	Latin-1
000	083	05/03	LATIN CAPITAL LETTER S	Latin-1
000	084	05/04	LATIN CAPITAL LETTER T	Latin-1
000	085	05/05	LATIN CAPITAL LETTER U	Latin-1
000	086	05/06	LATIN CAPITAL LETTER V	Latin-1
000	087	05/07	LATIN CAPITAL LETTER W	Latin-1
000	088	05/08	LATIN CAPITAL LETTER X	Latin-1
000	089	05/09	LATIN CAPITAL LETTER Y	Latin-1
000	090	05/10	LATIN CAPITAL LETTER Z	Latin-1
000	091	05/11	LEFT SQUARE BRACKET	Latin-1
000	092	05/12	REVERSE SOLIDUS	Latin-1
000	093	05/13	RIGHT SQUARE BRACKET	Latin-1
000	094	05/14	CIRCUMFLEX ACCENT	Latin-1
000	095	05/15	LOW LINE	Latin-1
000	096	06/00	GRAVE ACCENT	Latin-1
000	097	06/01	LATIN SMALL LETTER a	Latin-1
000	098	06/02	LATIN SMALL LETTER b	Latin-1
000	099	06/03	LATIN SMALL LETTER c	Latin-1
000	100	06/04	LATIN SMALL LETTER d	Latin-1
000	101	06/05	LATIN SMALL LETTER e	Latin-1
000	102	06/06	LATIN SMALL LETTER f	Latin-1
000	103	06/07	LATIN SMALL LETTER g	Latin-1
000	104	06/08	LATIN SMALL LETTER h	Latin-1
000	105	06/09	LATIN SMALL LETTER i	Latin-1
000	106	06/10	LATIN SMALL LETTER j	Latin-1
000	107	06/11	LATIN SMALL LETTER k	Latin-1
000	108	06/12	LATIN SMALL LETTER l	Latin-1

Byte 3	Byte 4	Code Pos	Name	Set
000	109	06/13	LATIN SMALL LETTER m	Latin-1
000	110	06/14	LATIN SMALL LETTER n	Latin-1
000	111	06/15	LATIN SMALL LETTER o	Latin-1
000	112	07/00	LATIN SMALL LETTER p	Latin-1
000	113	07/01	LATIN SMALL LETTER q	Latin-1
000	114	07/02	LATIN SMALL LETTER r	Latin-1
000	115	07/03	LATIN SMALL LETTER s	Latin-1
000	116	07/04	LATIN SMALL LETTER t	Latin-1
000	117	07/05	LATIN SMALL LETTER u	Latin-1
000	118	07/06	LATIN SMALL LETTER v	Latin-1
000	119	07/07	LATIN SMALL LETTER w	Latin-1
000	120	07/08	LATIN SMALL LETTER x	Latin-1
000	121	07/09	LATIN SMALL LETTER y	Latin-1
000	122	07/10	LATIN SMALL LETTER z	Latin-1
000	123	07/11	LEFT CURLY BRACKET	Latin-1
000	124	07/12	VERTICAL LINE	Latin-1
000	125	07/13	RIGHT CURLY BRACKET	Latin-1
000	126	07/14	TILDE	Latin-1
000	160	10/00	NO-BREAK SPACE	Latin-1
000	161	10/01	INVERTED EXCLAMATION MARK	Latin-1
000	162	10/02	CENT SIGN	Latin-1
000	163	10/03	POUND SIGN	Latin-1
000	164	10/04	CURRENCY SIGN	Latin-1
000	165	10/05	YEN SIGN	Latin-1
000	166	10/06	BROKEN VERTICAL BAR	Latin-1
000	167	10/07	SECTION SIGN	Latin-1
000	168	10/08	DIAERESIS	Latin-1
000	169	10/09	COPYRIGHT SIGN	Latin-1
000	170	10/10	FEMININE ORDINAL INDICATOR	Latin-1
000	171	10/11	LEFT ANGLE QUOTATION MARK	Latin-1
000	172	10/12	NOT SIGN	Latin-1
000	173	10/13	HYPHEN	Latin-1
000	174	10/14	REGISTERED TRADEMARK SIGN	Latin-1
000	175	10/15	MACRON	Latin-1
000	176	11/00	DEGREE SIGN, RING ABOVE	Latin-1
000	177	11/01	PLUS-MINUS SIGN	Latin-1
000	178	11/02	SUPERSCRIPT TWO	Latin-1
000	179	11/03	SUPERSCRIPT THREE	Latin-1
000	180	11/04	ACUTE ACCENT	Latin-1
000	181	11/05	MICRO SIGN	Latin-1
000	182	11/06	PARAGRAPH SIGN	Latin-1
000	183	11/07	MIDDLE DOT	Latin-1
000	184	11/08	CEDILLA	Latin-1
000	185	11/09	SUPERSCRIPT ONE	Latin-1
000	186	11/10	MASCULINE ORDINAL INDICATOR	Latin-1
000	187	11/11	RIGHT ANGLE QUOTATION MARK	Latin-1
000	188	11/12	VULGAR FRACTION ONE QUARTER	Latin-1
000	189	11/13	VULGAR FRACTION ONE HALF	Latin-1
000	190	11/14	VULGAR FRACTION THREE QUARTERS	Latin-1
000	191	11/15	INVERTED QUESTION MARK	Latin-1
000	192	12/00	LATIN CAPITAL LETTER A WITH GRAVE ACCENT	Latin-1
000	193	12/01	LATIN CAPITAL LETTER A WITH ACUTE ACCENT	Latin-1
000	194	12/02	LATIN CAPITAL LETTER A WITH CIRCUMFLEX ACCENT	Latin-1
000	195	12/03	LATIN CAPITAL LETTER A WITH TILDE	Latin-1
000	196	12/04	LATIN CAPITAL LETTER A WITH DIAERESIS	Latin-1
000	197	12/05	LATIN CAPITAL LETTER A WITH RING ABOVE	Latin-1
000	198	12/06	LATIN CAPITAL DIPHTHONG AE	Latin-1
000	199	12/07	LATIN CAPITAL LETTER C WITH CEDILLA	Latin-1
000	200	12/08	LATIN CAPITAL LETTER E WITH GRAVE ACCENT	Latin-1
000	201	12/09	LATIN CAPITAL LETTER E WITH ACUTE ACCENT	Latin-1
000	202	12/10	LATIN CAPITAL LETTER E WITH CIRCUMFLEX ACCENT	Latin-1

Byte 3	Byte 4	Code Pos	Name	Set
000	203	12/11	LATIN CAPITAL LETTER E WITH DIAERESIS	Latin-1
000	204	12/12	LATIN CAPITAL LETTER I WITH GRAVE ACCENT	Latin-1
000	205	12/13	LATIN CAPITAL LETTER I WITH ACUTE ACCENT	Latin-1
000	206	12/14	LATIN CAPITAL LETTER I WITH CIRCUMFLEX ACCENT	Latin-1
000	207	12/15	LATIN CAPITAL LETTER I WITH DIAERESIS	Latin-1
000	208	13/00	ICELANDIC CAPITAL LETTER ETH	Latin-1
000	209	13/01	LATIN CAPITAL LETTER N WITH TILDE	Latin-1
000	210	13/02	LATIN CAPITAL LETTER O WITH GRAVE ACCENT	Latin-1
000	211	13/03	LATIN CAPITAL LETTER O WITH ACUTE ACCENT	Latin-1
000	212	13/04	LATIN CAPITAL LETTER O WITH CIRCUMFLEX ACCENT	Latin-1
000	213	13/05	LATIN CAPITAL LETTER O WITH TILDE	Latin-1
000	214	13/06	LATIN CAPITAL LETTER O WITH DIAERESIS	Latin-1
000	215	13/07	MULTIPLICATION SIGN	Latin-1
000	216	13/08	LATIN CAPITAL LETTER O WITH OBLIQUE STROKE	Latin-1
000	217	13/09	LATIN CAPITAL LETTER U WITH GRAVE ACCENT	Latin-1
000	218	13/10	LATIN CAPITAL LETTER U WITH ACUTE ACCENT	Latin-1
000	219	13/11	LATIN CAPITAL LETTER U WITH CIRCUMFLEX ACCENT	Latin-1
000	220	13/12	LATIN CAPITAL LETTER U WITH DIAERESIS	Latin-1
000	221	13/13	LATIN CAPITAL LETTER Y WITH ACUTE ACCENT	Latin-1
000	222	13/14	ICELANDIC CAPITAL LETTER THORN	Latin-1
000	223	13/15	GERMAN SMALL LETTER SHARP s	Latin-1
000	224	14/00	LATIN SMALL LETTER a WITH GRAVE ACCENT	Latin-1
000	225	14/01	LATIN SMALL LETTER a WITH ACUTE ACCENT	Latin-1
000	226	14/02	LATIN SMALL LETTER a WITH CIRCUMFLEX ACCENT	Latin-1
000	227	14/03	LATIN SMALL LETTER a WITH TILDE	Latin-1
000	228	14/04	LATIN SMALL LETTER a WITH DIAERESIS	Latin-1
000	229	14/05	LATIN SMALL LETTER a WITH RING ABOVE	Latin-1
000	230	14/06	LATIN SMALL DIPHTHONG ae	Latin-1
000	231	14/07	LATIN SMALL LETTER c WITH CEDILLA	Latin-1
000	232	14/08	LATIN SMALL LETTER e WITH GRAVE ACCENT	Latin-1
000	233	14/09	LATIN SMALL LETTER e WITH ACUTE ACCENT	Latin-1
000	234	14/10	LATIN SMALL LETTER e WITH CIRCUMFLEX ACCENT	Latin-1
000	235	14/11	LATIN SMALL LETTER e WITH DIAERESIS	Latin-1
000	236	14/12	LATIN SMALL LETTER i WITH GRAVE ACCENT	Latin-1
000	237	14/13	LATIN SMALL LETTER i WITH ACUTE ACCENT	Latin-1
000	238	14/14	LATIN SMALL LETTER i WITH CIRCUMFLEX ACCENT	Latin-1
000	239	14/15	LATIN SMALL LETTER i WITH DIAERESIS	Latin-1
000	240	15/00	ICELANDIC SMALL LETTER ETH	Latin-1
000	241	15/01	LATIN SMALL LETTER n WITH TILDE	Latin-1
000	242	15/02	LATIN SMALL LETTER o WITH GRAVE ACCENT	Latin-1
000	243	15/03	LATIN SMALL LETTER o WITH ACUTE ACCENT	Latin-1
000	244	15/04	LATIN SMALL LETTER o WITH CIRCUMFLEX ACCENT	Latin-1
000	245	15/05	LATIN SMALL LETTER o WITH TILDE	Latin-1
000	246	15/06	LATIN SMALL LETTER o WITH DIAERESIS	Latin-1
000	247	15/07	DIVISION SIGN	Latin-1
000	248	15/08	LATIN SMALL LETTER o WITH OBLIQUE STROKE	Latin-1
000	249	15/09	LATIN SMALL LETTER u WITH GRAVE ACCENT	Latin-1
000	250	15/10	LATIN SMALL LETTER u WITH ACUTE ACCENT	Latin-1
000	251	15/11	LATIN SMALL LETTER u WITH CIRCUMFLEX ACCENT	Latin-1
000	252	15/12	LATIN SMALL LETTER u WITH DIAERESIS	Latin-1
000	253	15/13	LATIN SMALL LETTER y WITH ACUTE ACCENT	Latin-1
000	254	15/14	ICELANDIC SMALL LETTER THORN	Latin-1
000	255	15/15	LATIN SMALL LETTER y WITH DIAERESIS	Latin-1
001	161	10/01	LATIN CAPITAL LETTER A WITH OGONEK	Latin-2
001	162	10/02	BREVE	Latin-2
001	163	10/03	LATIN CAPITAL LETTER L WITH STROKE	Latin-2
001	165	10/05	LATIN CAPITAL LETTER L WITH CARON	Latin-2
001	166	10/06	LATIN CAPITAL LETTER S WITH ACUTE ACCENT	Latin-2
001	169	10/09	LATIN CAPITAL LETTER S WITH CARON	Latin-2

Byte 3	Byte 4	Code Pos	Name	Set
001	170	10/10	LATIN CAPITAL LETTER S WITH CEDILLA	Latin-2
001	171	10/11	LATIN CAPITAL LETTER T WITH CARON	Latin-2
001	172	10/12	LATIN CAPITAL LETTER Z WITH ACUTE ACCENT	Latin-2
001	174	10/14	LATIN CAPITAL LETTER Z WITH CARON	Latin-2
001	175	10/15	LATIN CAPITAL LETTER Z WITH DOT ABOVE	Latin-2
001	177	11/01	LATIN SMALL LETTER a WITH OGONEK	Latin-2
001	178	11/02	OGONEK	Latin-2
001	179	11/03	LATIN SMALL LETTER l WITH STROKE	Latin-2
001	181	11/05	LATIN SMALL LETTER l WITH CARON	Latin-2
001	182	11/06	LATIN SMALL LETTER s WITH ACUTE ACCENT	Latin-2
001	183	11/07	CARON	Latin-2
001	185	11/09	LATIN SMALL LETTER s WITH CARON	Latin-2
001	186	11/10	LATIN SMALL LETTER s WITH CEDILLA	Latin-2
001	187	11/11	LATIN SMALL LETTER t WITH CARON	Latin-2
001	188	11/12	LATIN SMALL LETTER z WITH ACUTE ACCENT	Latin-2
001	189	11/13	DOUBLE ACUTE ACCENT	Latin-2
001	190	11/14	LATIN SMALL LETTER z WITH CARON	Latin-2
001	191	11/15	LATIN SMALL LETTER z WITH DOT ABOVE	Latin-2
001	192	12/00	LATIN CAPITAL LETTER R WITH ACUTE ACCENT	Latin-2
001	195	12/03	LATIN CAPITAL LETTER A WITH BREVE	Latin-2
001	197	12/05	LATIN CAPITAL LETTER L WITH ACUTE ACCENT	Latin-2
001	198	12/06	LATIN CAPITAL LETTER C WITH ACUTE ACCENT	Latin-2
001	200	12/08	LATIN CAPITAL LETTER C WITH CARON	Latin-2
001	202	12/10	LATIN CAPITAL LETTER E WITH OGONEK	Latin-2
001	204	12/12	LATIN CAPITAL LETTER E WITH CARON	Latin-2
001	207	12/15	LATIN CAPITAL LETTER D WITH CARON	Latin-2
001	208	13/00	LATIN CAPITAL LETTER D WITH STROKE	Latin-2
001	209	13/01	LATIN CAPITAL LETTER N WITH ACUTE ACCENT	Latin-2
001	210	13/02	LATIN CAPITAL LETTER N WITH CARON	Latin-2
001	213	13/05	LATIN CAPITAL LETTER O WITH DOUBLE ACUTE ACCENT	Latin-2
001	216	13/08	LATIN CAPITAL LETTER R WITH CARON	Latin-2
001	217	13/09	LATIN CAPITAL LETTER U WITH RING ABOVE	Latin-2
001	219	13/11	LATIN CAPITAL LETTER U WITH DOUBLE ACUTE ACCENT	Latin-2
001	222	13/14	LATIN CAPITAL LETTER T WITH CEDILLA	Latin-2
001	224	14/00	LATIN SMALL LETTER r WITH ACUTE ACCENT	Latin-2
001	227	14/03	LATIN SMALL LETTER a WITH BREVE	Latin-2
001	229	14/05	LATIN SMALL LETTER l WITH ACUTE ACCENT	Latin-2
001	230	14/06	LATIN SMALL LETTER c WITH ACUTE ACCENT	Latin-2
001	232	14/08	LATIN SMALL LETTER c WITH CARON	Latin-2
001	234	14/10	LATIN SMALL LETTER e WITH OGONEK	Latin-2
001	236	14/12	LATIN SMALL LETTER e WITH CARON	Latin-2
001	239	14/15	LATIN SMALL LETTER d WITH CARON	Latin-2
001	240	15/00	LATIN SMALL LETTER d WITH STROKE	Latin-2
001	241	15/01	LATIN SMALL LETTER n WITH ACUTE ACCENT	Latin-2
001	242	15/02	LATIN SMALL LETTER n WITH CARON	Latin-2
001	245	15/05	LATIN SMALL LETTER o WITH DOUBLE ACUTE ACCENT	Latin-2
001	248	15/08	LATIN SMALL LETTER r WITH CARON	Latin-2
001	249	15/09	LATIN SMALL LETTER u WITH RING ABOVE	Latin-2
001	251	15/11	LATIN SMALL LETTER u WITH DOUBLE ACUTE ACCENT	Latin-2
001	254	15/14	LATIN SMALL LETTER t WITH CEDILLA	Latin-2
001	255	15/15	DOT ABOVE	Latin-2
002	161	10/01	LATIN CAPITAL LETTER H WITH STROKE	Latin-3
002	166	10/06	LATIN CAPITAL LETTER H WITH CIRCUMFLEX ACCENT	Latin-3
002	169	10/09	LATIN CAPITAL LETTER I WITH DOT ABOVE	Latin-3
002	171	10/11	LATIN CAPITAL LETTER G WITH BREVE	Latin-3
002	172	10/12	LATIN CAPITAL LETTER J WITH CIRCUMFLEX ACCENT	Latin-3
002	177	11/01	LATIN SMALL LETTER h WITH STROKE	Latin-3
002	182	11/06	LATIN SMALL LETTER h WITH CIRCUMFLEX ACCENT	Latin-3
002	185	11/09	SMALL DOTLESS LETTER i	Latin-3

Byte 3	Byte 4	Code Pos	Name	Set
002	187	11/11	LATIN SMALL LETTER g WITH BREVE	Latin-3
002	188	11/12	LATIN SMALL LETTER j WITH CIRCUMFLEX ACCENT	Latin-3
002	197	12/05	LATIN CAPITAL LETTER C WITH DOT ABOVE	Latin-3
002	198	12/06	LATIN CAPITAL LETTER C WITH CIRCUMFLEX ACCENT	Latin-3
002	213	13/05	LATIN CAPITAL LETTER G WITH DOT ABOVE	Latin-3
002	216	13/08	LATIN CAPITAL LETTER G WITH CIRCUMFLEX ACCENT	Latin-3
002	221	13/13	LATIN CAPITAL LETTER U WITH BREVE	Latin-3
002	222	13/14	LATIN CAPITAL LETTER S WITH CIRCUMFLEX ACCENT	Latin-3
002	229	14/05	LATIN SMALL LETTER c WITH DOT ABOVE	Latin-3
002	230	14/06	LATIN SMALL LETTER c WITH CIRCUMFLEX ACCENT	Latin-3
002	245	15/05	LATIN SMALL LETTER g WITH DOT ABOVE	Latin-3
002	248	15/08	LATIN SMALL LETTER g WITH CIRCUMFLEX ACCENT	Latin-3
002	253	15/13	LATIN SMALL LETTER u WITH BREVE	Latin-3
002	254	15/14	LATIN SMALL LETTER s WITH CIRCUMFLEX ACCENT	Latin-3
003	162	10/02	SMALL GREENLANDIC LETTER KRA	Latin-4
003	163	10/03	LATIN CAPITAL LETTER R WITH CEDILLA	Latin-4
003	165	10/05	LATIN CAPITAL LETTER I WITH TILDE	Latin-4
003	166	10/06	LATIN CAPITAL LETTER L WITH CEDILLA	Latin-4
003	170	10/10	LATIN CAPITAL LETTER E WITH MACRON	Latin-4
003	171	10/11	LATIN CAPITAL LETTER G WITH CEDILLA	Latin-4
003	172	10/12	LATIN CAPITAL LETTER T WITH OBLIQUE STROKE	Latin-4
003	179	11/03	LATIN SMALL LETTER r WITH CEDILLA	Latin-4
003	181	11/05	LATIN SMALL LETTER i WITH TILDE	Latin-4
003	182	11/06	LATIN SMALL LETTER l WITH CEDILLA	Latin-4
003	186	11/10	LATIN SMALL LETTER e WITH MACRON	Latin-4
003	187	11/11	LATIN SMALL LETTER g WITH CEDILLA ABOVE	Latin-4
003	188	11/12	LATIN SMALL LETTER t WITH OBLIQUE STROKE	Latin-4
003	189	11/13	LAPPISH CAPITAL LETTER ENG	Latin-4
003	191	11/15	LAPPISH SMALL LETTER ENG	Latin-4
003	192	12/00	LATIN CAPITAL LETTER A WITH MACRON	Latin-4
003	199	12/07	LATIN CAPITAL LETTER I WITH OGONEK	Latin-4
003	204	12/12	LATIN CAPITAL LETTER E WITH DOT ABOVE	Latin-4
003	207	12/15	LATIN CAPITAL LETTER I WITH MACRON	Latin-4
003	209	13/01	LATIN CAPITAL LETTER N WITH CEDILLA	Latin-4
003	210	13/02	LATIN CAPITAL LETTER O WITH MACRON	Latin-4
003	211	13/03	LATIN CAPITAL LETTER K WITH CEDILLA	Latin-4
003	217	13/09	LATIN CAPITAL LETTER U WITH OGONEK	Latin-4
003	221	13/13	LATIN CAPITAL LETTER U WITH TILDE	Latin-4
003	222	13/14	LATIN CAPITAL LETTER U WITH MACRON	Latin-4
003	224	14/00	LATIN SMALL LETTER a WITH MACRON	Latin-4
003	231	14/07	LATIN SMALL LETTER i WITH OGONEK	Latin-4
003	236	14/12	LATIN SMALL LETTER e WITH DOT ABOVE	Latin-4
003	239	14/15	LATIN SMALL LETTER i WITH MACRON	Latin-4
003	241	15/01	LATIN SMALL LETTER n WITH CEDILLA	Latin-4
003	242	15/02	LATIN SMALL LETTER o WITH MACRON	Latin-4
003	243	15/03	LATIN SMALL LETTER k WITH CEDILLA	Latin-4
003	249	15/09	LATIN SMALL LETTER u WITH OGONEK	Latin-4
003	253	15/13	LATIN SMALL LETTER u WITH TILDE	Latin-4
003	254	15/14	LATIN SMALL LETTER u WITH MACRON	Latin-4
004	126	07/14	OVERLINE	Kana
004	161	10/01	KANA FULL STOP	Kana
004	162	10/02	KANA OPENING BRACKET	Kana
004	163	10/03	KANA CLOSING BRACKET	Kana
004	164	10/04	KANA COMMA	Kana
004	165	10/05	KANA CONJUNCTIVE	Kana
004	166	10/06	KANA LETTER WO	Kana
004	167	10/07	KANA LETTER SMALL A	Kana

Byte 3	Byte 4	Code Pos	Name	Set
004	168	10/08	KANA LETTER SMALL I	Kana
004	169	10/09	KANA LETTER SMALL U	Kana
004	170	10/10	KANA LETTER SMALL E	Kana
004	171	10/11	KANA LETTER SMALL O	Kana
004	172	10/12	KANA LETTER SMALL YA	Kana
004	173	10/13	KANA LETTER SMALL YU	Kana
004	174	10/14	KANA LETTER SMALL YO	Kana
004	175	10/15	KANA LETTER SMALL TSU	Kana
004	176	11/00	PROLONGED SOUND SYMBOL	Kana
004	177	11/01	KANA LETTER A	Kana
004	178	11/02	KANA LETTER I	Kana
004	179	11/03	KANA LETTER U	Kana
004	180	11/04	KANA LETTER E	Kana
004	181	11/05	KANA LETTER O	Kana
004	182	11/06	KANA LETTER KA	Kana
004	183	11/07	KANA LETTER KI	Kana
004	184	11/08	KANA LETTER KU	Kana
004	185	11/09	KANA LETTER KE	Kana
004	186	11/10	KANA LETTER KO	Kana
004	187	11/11	KANA LETTER SA	Kana
004	188	11/12	KANA LETTER SHI	Kana
004	189	11/13	KANA LETTER SU	Kana
004	190	11/14	KANA LETTER SE	Kana
004	191	11/15	KANA LETTER SO	Kana
004	192	12/00	KANA LETTER TA	Kana
004	193	12/01	KANA LETTER CHI	Kana
004	194	12/02	KANA LETTER TSU	Kana
004	195	12/03	KANA LETTER TE	Kana
004	196	12/04	KANA LETTER TO	Kana
004	197	12/05	KANA LETTER NA	Kana
004	198	12/06	KANA LETTER NI	Kana
004	199	12/07	KANA LETTER NU	Kana
004	200	12/08	KANA LETTER NE	Kana
004	201	12/09	KANA LETTER NO	Kana
004	202	12/10	KANA LETTER HA	Kana
004	203	12/11	KANA LETTER HI	Kana
004	204	12/12	KANA LETTER FU	Kana
004	205	12/13	KANA LETTER HE	Kana
004	206	12/14	KANA LETTER HO	Kana
004	207	12/15	KANA LETTER MA	Kana
004	208	13/00	KANA LETTER MI	Kana
004	209	13/01	KANA LETTER MU	Kana
004	210	13/02	KANA LETTER ME	Kana
004	211	13/03	KANA LETTER MO	Kana
004	212	13/04	KANA LETTER YA	Kana
004	213	13/05	KANA LETTER YU	Kana
004	214	13/06	KANA LETTER YO	Kana
004	215	13/07	KANA LETTER RA	Kana
004	216	13/08	KANA LETTER RI	Kana
004	217	13/09	KANA LETTER RU	Kana
004	218	13/10	KANA LETTER RE	Kana
004	219	13/11	KANA LETTER RO	Kana
004	220	13/12	KANA LETTER WA	Kana
004	221	13/13	KANA LETTER N	Kana
004	222	13/14	VOICED SOUND SYMBOL	Kana
004	223	13/15	SEMIVOICED SOUND SYMBOL	Kana
005	172	10/12	ARABIC COMMA	Arabic
005	187	11/11	ARABIC SEMICOLON	Arabic
005	191	11/15	ARABIC QUESTION MARK	Arabic

Byte 3	Byte 4	Code Pos	Name	Set
005	193	12/01	ARABIC LETTER HAMZA	Arabic
005	194	12/02	ARABIC LETTER MADDA ON ALEF	Arabic
005	195	12/03	ARABIC LETTER HAMZA ON ALEF	Arabic
005	196	12/04	ARABIC LETTER HAMZA ON WAW	Arabic
005	197	12/05	ARABIC LETTER HAMZA UNDER ALEF	Arabic
005	198	12/06	ARABIC LETTER HAMZA ON YEH	Arabic
005	199	12/07	ARABIC LETTER ALEF	Arabic
005	200	12/08	ARABIC LETTER BEH	Arabic
005	201	12/09	ARABIC LETTER TEH MARBUTA	Arabic
005	202	12/10	ARABIC LETTER TEH	Arabic
005	203	12/11	ARABIC LETTER THEH	Arabic
005	204	12/12	ARABIC LETTER JEEM	Arabic
005	205	12/13	ARABIC LETTER HAH	Arabic
005	206	12/14	ARABIC LETTER KHAH	Arabic
005	207	12/15	ARABIC LETTER DAL	Arabic
005	208	13/00	ARABIC LETTER THAL	Arabic
005	209	13/01	ARABIC LETTER RA	Arabic
005	210	13/02	ARABIC LETTER ZAIN	Arabic
005	211	13/03	ARABIC LETTER SEEN	Arabic
005	212	13/04	ARABIC LETTER SHEEN	Arabic
005	213	13/05	ARABIC LETTER SAD	Arabic
005	214	13/06	ARABIC LETTER DAD	Arabic
005	215	13/07	ARABIC LETTER TAH	Arabic
005	216	13/08	ARABIC LETTER ZAH	Arabic
005	217	13/09	ARABIC LETTER AIN	Arabic
005	218	13/10	ARABIC LETTER GHAIN	Arabic
005	224	14/00	ARABIC LETTER TATWEEL	Arabic
005	225	14/01	ARABIC LETTER FEH	Arabic
005	226	14/02	ARABIC LETTER QAF	Arabic
005	227	14/03	ARABIC LETTER KAF	Arabic
005	228	14/04	ARABIC LETTER LAM	Arabic
005	229	14/05	ARABIC LETTER MEEM	Arabic
005	230	14/06	ARABIC LETTER NOON	Arabic
005	231	14/07	ARABIC LETTER HA	Arabic
005	232	14/08	ARABIC LETTER WAW	Arabic
005	233	14/09	ARABIC LETTER ALEF MAKSURA	Arabic
005	234	14/10	ARABIC LETTER YEH	Arabic
005	235	14/11	ARABIC LETTER FATHATAN	Arabic
005	236	14/12	ARABIC LETTER DAMMATAN	Arabic
005	237	14/13	ARABIC LETTER KASRATAN	Arabic
005	238	14/14	ARABIC LETTER FATHA	Arabic
005	239	14/15	ARABIC LETTER DAMMA	Arabic
005	240	15/00	ARABIC LETTER KASRA	Arabic
005	241	15/01	ARABIC LETTER SHADDA	Arabic
005	242	15/02	ARABIC LETTER SUKUN	Arabic
006	161	10/01	SERBOCROATION CYRILLIC SMALL LETTER DJE	Cyrillic
006	162	10/02	MACEDONIAN CYRILLIC SMALL LETTER GJE	Cyrillic
006	163	10/03	CYRILLIC SMALL LETTER IO	Cyrillic
006	164	10/04	UKRAINIAN CYRILLIC SMALL LETTER IE	Cyrillic
006	165	10/05	MACEDONIAN SMALL LETTER DSE	Cyrillic
006	166	10/06	BYELORUSSIAN/UKRAINIAN CYRILLIC SMALL LETTER I	Cyrillic
006	167	10/07	UKRAINIAN SMALL LETTER YI	Cyrillic
006	168	10/08	CYRILLIC SMALL LETTER JE	Cyrillic
006	169	10/09	CYRILLIC SMALL LETTER LJE	Cyrillic
006	170	10/10	CYRILLIC SMALL LETTER NJE	Cyrillic
006	171	10/11	SERBIAN SMALL LETTER TSHE	Cyrillic
006	172	10/12	MACEDONIAN CYRILLIC SMALL LETTER KJE	Cyrillic
006	174	10/14	BYELORUSSIAN SMALL LETTER SHORT U	Cyrillic
006	175	10/15	CYRILLIC SMALL LETTER DZHE	Cyrillic

Byte 3	Byte 4	Code Pos	Name	Set
006	176	11/00	NUMERO SIGN	Cyrillic
006	177	11/01	SERBOCROATIAN CYRILLIC CAPITAL LETTER DJE	Cyrillic
006	178	11/02	MACEDONIAN CYRILLIC CAPITAL LETTER GJE	Cyrillic
006	179	11/03	CYRILLIC CAPITAL LETTER IO	Cyrillic
006	180	11/04	UKRAINIAN CYRILLIC CAPITAL LETTER IE	Cyrillic
006	181	11/05	MACEDONIAN CAPITAL LETTER DSE	Cyrillic
006	182	11/06	BYELORUSSIAN/UKRAINIAN CYRILLIC CAPITAL LETTER I	Cyrillic
006	183	11/07	UKRAINIAN CAPITAL LETTER YI	Cyrillic
006	184	11/08	CYRILLIC CAPITAL LETTER JE	Cyrillic
006	185	11/09	CYRILLIC CAPITAL LETTER LJE	Cyrillic
006	186	11/10	CYRILLIC CAPITAL LETTER NJE	Cyrillic
006	187	11/11	SERBIAN CAPITAL LETTER TSHE	Cyrillic
006	188	11/12	MACEDONIAN CYRILLIC CAPITAL LETTER KJE	Cyrillic
006	190	11/14	BYELORUSSIAN CAPITAL LETTER SHORT U	Cyrillic
006	191	11/15	CYRILLIC CAPITAL LETTER DZHE	Cyrillic
006	192	12/00	CYRILLIC SMALL LETTER YU	Cyrillic
006	193	12/01	CYRILLIC SMALL LETTER A	Cyrillic
006	194	12/02	CYRILLIC SMALL LETTER BE	Cyrillic
006	195	12/03	CYRILLIC SMALL LETTER TSE	Cyrillic
006	196	12/04	CYRILLIC SMALL LETTER DE	Cyrillic
006	197	12/05	CYRILLIC SMALL LETTER IE	Cyrillic
006	198	12/06	CYRILLIC SMALL LETTER EF	Cyrillic
006	199	12/07	CYRILLIC SMALL LETTER GHE	Cyrillic
006	200	12/08	CYRILLIC SMALL LETTER HA	Cyrillic
006	201	12/09	CYRILLIC SMALL LETTER I	Cyrillic
006	202	12/10	CYRILLIC SMALL LETTER SHORT I	Cyrillic
006	203	12/11	CYRILLIC SMALL LETTER KA	Cyrillic
006	204	12/12	CYRILLIC SMALL LETTER EL	Cyrillic
006	205	12/13	CYRILLIC SMALL LETTER EM	Cyrillic
006	206	12/14	CYRILLIC SMALL LETTER EN	Cyrillic
006	207	12/15	CYRILLIC SMALL LETTER O	Cyrillic
006	208	13/00	CYRILLIC SMALL LETTER PE	Cyrillic
006	209	13/01	CYRILLIC SMALL LETTER YA	Cyrillic
006	210	13/02	CYRILLIC SMALL LETTER ER	Cyrillic
006	211	13/03	CYRILLIC SMALL LETTER ES	Cyrillic
006	212	13/04	CYRILLIC SMALL LETTER TE	Cyrillic
006	213	13/05	CYRILLIC SMALL LETTER U	Cyrillic
006	214	13/06	CYRILLIC SMALL LETTER ZHE	Cyrillic
006	215	13/07	CYRILLIC SMALL LETTER VE	Cyrillic
006	216	13/08	CYRILLIC SMALL SOFT SIGN	Cyrillic
006	217	13/09	CYRILLIC SMALL LETTER YERU	Cyrillic
006	218	13/10	CYRILLIC SMALL LETTER ZE	Cyrillic
006	219	13/11	CYRILLIC SMALL LETTER SHA	Cyrillic
006	220	13/12	CYRILLIC SMALL LETTER E	Cyrillic
006	221	13/13	CYRILLIC SMALL LETTER SHCHA	Cyrillic
006	222	13/14	CYRILLIC SMALL LETTER CHE	Cyrillic
006	223	13/15	CYRILLIC SMALL HARD SIGN	Cyrillic
006	224	14/00	CYRILLIC CAPITAL LETTER YU	Cyrillic
006	225	14/01	CYRILLIC CAPITAL LETTER A	Cyrillic
006	226	14/02	CYRILLIC CAPITAL LETTER BE	Cyrillic
006	227	14/03	CYRILLIC CAPITAL LETTER TSE	Cyrillic
006	228	14/04	CYRILLIC CAPITAL LETTER DE	Cyrillic
006	229	14/05	CYRILLIC CAPITAL LETTER IE	Cyrillic
006	230	14/06	CYRILLIC CAPITAL LETTER EF	Cyrillic
006	231	14/07	CYRILLIC CAPITAL LETTER GHE	Cyrillic
006	232	14/08	CYRILLIC CAPITAL LETTER HA	Cyrillic
006	233	14/09	CYRILLIC CAPITAL LETTER I	Cyrillic
006	234	14/10	CYRILLIC CAPITAL LETTER SHORT I	Cyrillic
006	235	14/11	CYRILLIC CAPITAL LETTER KA	Cyrillic
006	236	14/12	CYRILLIC CAPITAL LETTER EL	Cyrillic
006	237	14/13	CYRILLIC CAPITAL LETTER EM	Cyrillic

Byte 3	Byte 4	Code Pos	Name	Set
006	238	14/14	CYRILLIC CAPITAL LETTER EN	Cyrillic
006	239	14/15	CYRILLIC CAPITAL LETTER O	Cyrillic
006	240	15/00	CYRILLIC CAPITAL LETTER PE	Cyrillic
006	241	15/01	CYRILLIC CAPITAL LETTER YA	Cyrillic
006	242	15/02	CYRILLIC CAPITAL LETTER ER	Cyrillic
006	243	15/03	CYRILLIC CAPITAL LETTER ES	Cyrillic
006	244	15/04	CYRILLIC CAPITAL LETTER TE	Cyrillic
006	245	15/05	CYRILLIC CAPITAL LETTER U	Cyrillic
006	246	15/06	CYRILLIC CAPITAL LETTER ZHE	Cyrillic
006	247	15/07	CYRILLIC CAPITAL LETTER VE	Cyrillic
006	248	15/08	CYRILLIC CAPITAL SOFT SIGN	Cyrillic
006	249	15/09	CYRILLIC CAPITAL LETTER YERU	Cyrillic
006	250	15/10	CYRILLIC CAPITAL LETTER ZE	Cyrillic
006	251	15/11	CYRILLIC CAPITAL LETTER SHA	Cyrillic
006	252	15/12	CYRILLIC CAPITAL LETTER E	Cyrillic
006	253	15/13	CYRILLIC CAPITAL LETTER SHCHA	Cyrillic
006	254	15/14	CYRILLIC CAPITAL LETTER CHE	Cyrillic
006	255	15/15	CYRILLIC CAPITAL HARD SIGN	Cyrillic
007	161	10/01	GREEK CAPITAL LETTER ALPHA WITH ACCENT	Greek
007	162	10/02	GREEK CAPITAL LETTER EPSILON WITH ACCENT	Greek
007	163	10/03	GREEK CAPITAL LETTER ETA WITH ACCENT	Greek
007	164	10/04	GREEK CAPITAL LETTER IOTA WITH ACCENT	Greek
007	165	10/05	GREEK CAPITAL LETTER IOTA WITH DIAERESIS	Greek
007	167	10/07	GREEK CAPITAL LETTER OMICRON WITH ACCENT	Greek
007	168	10/08	GREEK CAPITAL LETTER UPSILON WITH ACCENT	Greek
007	169	10/09	GREEK CAPITAL LETTER UPSILON WITH DIAERESIS	Greek
007	171	10/11	GREEK CAPITAL LETTER OMEGA WITH ACCENT	Greek
007	174	10/14	DIAERESIS AND ACCENT	Greek
007	175	10/15	HORIZONTAL BAR	Greek
007	177	11/01	GREEK SMALL LETTER ALPHA WITH ACCENT	Greek
007	178	11/02	GREEK SMALL LETTER EPSILON WITH ACCENT	Greek
007	179	11/03	GREEK SMALL LETTER ETA WITH ACCENT	Greek
007	180	11/04	GREEK SMALL LETTER IOTA WITH ACCENT	Greek
007	181	11/05	GREEK SMALL LETTER IOTA WITH DIAERESIS	Greek
007	182	11/06	GREEK SMALL LETTER IOTA WITH ACCENT+DIAERESIS	Greek
007	183	11/07	GREEK SMALL LETTER OMICRON WITH ACCENT	Greek
007	184	11/08	GREEK SMALL LETTER UPSILON WITH ACCENT	Greek
007	185	11/09	GREEK SMALL LETTER UPSILON WITH DIAERESIS	Greek
007	186	11/10	GREEK SMALL LETTER UPSILON WITH ACCENT+DIAERESIS	Greek
007	187	11/11	GREEK SMALL LETTER OMEGA WITH ACCENT	Greek
007	193	12/01	GREEK CAPITAL LETTER ALPHA	Greek
007	194	12/02	GREEK CAPITAL LETTER BETA	Greek
007	195	12/03	GREEK CAPITAL LETTER GAMMA	Greek
007	196	12/04	GREEK CAPITAL LETTER DELTA	Greek
007	197	12/05	GREEK CAPITAL LETTER EPSILON	Greek
007	198	12/06	GREEK CAPITAL LETTER ZETA	Greek
007	199	12/07	GREEK CAPITAL LETTER ETA	Greek
007	200	12/08	GREEK CAPITAL LETTER THETA	Greek
007	201	12/09	GREEK CAPITAL LETTER IOTA	Greek
007	202	12/10	GREEK CAPITAL LETTER KAPPA	Greek
007	203	12/11	GREEK CAPITAL LETTER LAMDA	Greek
007	204	12/12	GREEK CAPITAL LETTER MU	Greek
007	205	12/13	GREEK CAPITAL LETTER NU	Greek
007	206	12/14	GREEK CAPITAL LETTER XI	Greek
007	207	12/15	GREEK CAPITAL LETTER OMICRON	Greek
007	208	13/00	GREEK CAPITAL LETTER PI	Greek
007	209	13/01	GREEK CAPITAL LETTER RHO	Greek
007	210	13/02	GREEK CAPITAL LETTER SIGMA	Greek
007	212	13/04	GREEK CAPITAL LETTER TAU	Greek

Byte 3	Byte 4	Code Pos	Name	Set
007	213	13/05	GREEK CAPITAL LETTER UPSILON	Greek
007	214	13/06	GREEK CAPITAL LETTER PHI	Greek
007	215	13/07	GREEK CAPITAL LETTER CHI	Greek
007	216	13/08	GREEK CAPITAL LETTER PSI	Greek
007	217	13/09	GREEK CAPITAL LETTER OMEGA	Greek
007	225	14/01	GREEK SMALL LETTER ALPHA	Greek
007	226	14/02	GREEK SMALL LETTER BETA	Greek
007	227	14/03	GREEK SMALL LETTER GAMMA	Greek
007	228	14/04	GREEK SMALL LETTER DELTA	Greek
007	229	14/05	GREEK SMALL LETTER EPSILON	Greek
007	230	14/06	GREEK SMALL LETTER ZETA	Greek
007	231	14/07	GREEK SMALL LETTER ETA	Greek
007	232	14/08	GREEK SMALL LETTER THETA	Greek
007	233	14/09	GREEK SMALL LETTER IOTA	Greek
007	234	14/10	GREEK SMALL LETTER KAPPA	Greek
007	235	14/11	GREEK SMALL LETTER LAMDA	Greek
007	236	14/12	GREEK SMALL LETTER MU	Greek
007	237	14/13	GREEK SMALL LETTER NU	Greek
007	238	14/14	GREEK SMALL LETTER XI	Greek
007	239	14/15	GREEK SMALL LETTER OMICRON	Greek
007	240	15/00	GREEK SMALL LETTER PI	Greek
007	241	15/01	GREEK SMALL LETTER RHO	Greek
007	242	15/02	GREEK SMALL LETTER SIGMA	Greek
007	243	15/03	GREEK SMALL LETTER FINAL SMALL SIGMA	Greek
007	244	15/04	GREEK SMALL LETTER TAU	Greek
007	245	15/05	GREEK SMALL LETTER UPSILON	Greek
007	246	15/06	GREEK SMALL LETTER PHI	Greek
007	247	15/07	GREEK SMALL LETTER CHI	Greek
007	248	15/08	GREEK SMALL LETTER PSI	Greek
007	249	15/09	GREEK SMALL LETTER OMEGA	Greek
008	161	10/01	LEFT RADICAL	Technical
008	162	10/02	TOP LEFT RADICAL	Technical
008	163	10/03	HORIZONTAL CONNECTOR	Technical
008	164	10/04	TOP INTEGRAL	Technical
008	165	10/05	BOTTOM INTEGRAL	Technical
008	166	10/06	VERTICAL CONNECTOR	Technical
008	167	10/07	TOP LEFT SQUARE BRACKET	Technical
008	168	10/08	BOTTOM LEFT SQUARE BRACKET	Technical
008	169	10/09	TOP RIGHT SQUARE BRACKET	Technical
008	170	10/10	BOTTOM RIGHT SQUARE BRACKET	Technical
008	171	10/11	TOP LEFT PARENTHESIS	Technical
008	172	10/12	BOTTOM LEFT PARENTHESIS	Technical
008	173	10/13	TOP RIGHT PARENTHESIS	Technical
008	174	10/14	BOTTOM RIGHT PARENTHESIS	Technical
008	175	10/15	LEFT MIDDLE CURLY BRACE	Technical
008	176	11/00	RIGHT MIDDLE CURLY BRACE	Technical
008	177	11/01	TOP LEFT SUMMATION	Technical
008	178	11/02	BOTTOM LEFT SUMMATION	Technical
008	179	11/03	TOP VERTICAL SUMMATION CONNECTOR	Technical
008	180	11/04	BOTTOM VERTICAL SUMMATION CONNECTOR	Technical
008	181	11/05	TOP RIGHT SUMMATION	Technical
008	182	11/06	BOTTOM RIGHT SUMMATION	Technical
008	183	11/07	RIGHT MIDDLE SUMMATION	Technical
008	188	11/12	LESS THAN OR EQUAL SIGN	Technical
008	189	11/13	NOT EQUAL SIGN	Technical
008	190	11/14	GREATER THAN OR EQUAL SIGN	Technical
008	191	11/15	INTEGRAL	Technical
008	192	12/00	THEREFORE	Technical
008	193	12/01	VARIATION, PROPORTIONAL TO	Technical

Byte 3	Byte 4	Code Pos	Name	Set
008	194	12/02	INFINITY	Technical
008	197	12/05	NABLA, DEL	Technical
008	200	12/08	IS APPROXIMATE TO	Technical
008	201	12/09	SIMILAR OR EQUAL TO	Technical
008	205	12/13	IF AND ONLY IF	Technical
008	206	12/14	IMPLIES	Technical
008	207	12/15	IDENTICAL TO	Technical
008	214	13/06	RADICAL	Technical
008	218	13/10	IS INCLUDED IN	Technical
008	219	13/11	INCLUDES	Technical
008	220	13/12	INTERSECTION	Technical
008	221	13/13	UNION	Technical
008	222	13/14	LOGICAL AND	Technical
008	223	13/15	LOGICAL OR	Technical
008	239	14/15	PARTIAL DERIVATIVE	Technical
008	246	15/06	FUNCTION	Technical
008	251	15/11	LEFT ARROW	Technical
008	252	15/12	UPWARD ARROW	Technical
008	253	15/13	RIGHT ARROW	Technical
008	254	15/14	DOWNWARD ARROW	Technical
009	223	13/15	BLANK	Special
009	224	14/00	SOLID DIAMOND	Special
009	225	14/01	CHECKERBOARD	Special
009	226	14/02	‘‘HT’’	Special
009	227	14/03	‘‘FF’’	Special
009	228	14/04	‘‘CR’’	Special
009	229	14/05	‘‘LF’’	Special
009	232	14/08	‘‘NL’’	Special
009	233	14/09	‘‘VT’’	Special
009	234	14/10	LOWER-RIGHT CORNER	Special
009	235	14/11	UPPER-RIGHT CORNER	Special
009	236	14/12	UPPER-LEFT CORNER	Special
009	237	14/13	LOWER-LEFT CORNER	Special
009	238	14/14	CROSSING-LINES	Special
009	239	14/15	HORIZONTAL LINE, SCAN 1	Special
009	240	15/00	HORIZONTAL LINE, SCAN 3	Special
009	241	15/01	HORIZONTAL LINE, SCAN 5	Special
009	242	15/02	HORIZONTAL LINE, SCAN 7	Special
009	243	15/03	HORIZONTAL LINE, SCAN 9	Special
009	244	15/04	LEFT ‘‘T’’	Special
009	245	15/05	RIGHT ‘‘T’’	Special
009	246	15/06	BOTTOM ‘‘T’’	Special
009	247	15/07	TOP ‘‘T’’	Special
009	248	15/08	VERTICAL BAR	Special
010	161	10/01	EM SPACE	Publish
010	162	10/02	EN SPACE	Publish
010	163	10/03	3/EM SPACE	Publish
010	164	10/04	4/EM SPACE	Publish
010	165	10/05	DIGIT SPACE	Publish
010	166	10/06	PUNCTUATION SPACE	Publish
010	167	10/07	THIN SPACE	Publish
010	168	10/08	HAIR SPACE	Publish
010	169	10/09	EM DASH	Publish
010	170	10/10	EN DASH	Publish
010	172	10/12	SIGNIFICANT BLANK SYMBOL	Publish
010	174	10/14	ELLIPSIS	Publish
010	175	10/15	DOUBLE BASELINE DOT	Publish

Byte 3	Byte 4	Code Pos	Name	Set
010	176	11/00	VULGAR FRACTION ONE THIRD	Publish
010	177	11/01	VULGAR FRACTION TWO THIRDS	Publish
010	178	11/02	VULGAR FRACTION ONE FIFTH	Publish
010	179	11/03	VULGAR FRACTION TWO FIFTHS	Publish
010	180	11/04	VULGAR FRACTION THREE FIFTHS	Publish
010	181	11/05	VULGAR FRACTION FOUR FIFTHS	Publish
010	182	11/06	VULGAR FRACTION ONE SIXTH	Publish
010	183	11/07	VULGAR FRACTION FIVE SIXTHS	Publish
010	184	11/08	CARE OF	Publish
010	187	11/11	FIGURE DASH	Publish
010	188	11/12	LEFT ANGLE BRACKET	Publish
010	189	11/13	DECIMAL POINT	Publish
010	190	11/14	RIGHT ANGLE BRACKET	Publish
010	191	11/15	MARKER	Publish
010	195	12/03	VULGAR FRACTION ONE EIGHTH	Publish
010	196	12/04	VULGAR FRACTION THREE EIGHTHS	Publish
010	197	12/05	VULGAR FRACTION FIVE EIGHTHS	Publish
010	198	12/06	VULGAR FRACTION SEVEN EIGHTHS	Publish
010	201	12/09	TRADEMARK SIGN	Publish
010	202	12/10	SIGNATURE MARK	Publish
010	203	12/11	TRADEMARK SIGN IN CIRCLE	Publish
010	204	12/12	LEFT OPEN TRIANGLE	Publish
010	205	12/13	RIGHT OPEN TRIANGLE	Publish
010	206	12/14	EM OPEN CIRCLE	Publish
010	207	12/15	EM OPEN RECTANGLE	Publish
010	208	13/00	LEFT SINGLE QUOTATION MARK	Publish
010	209	13/01	RIGHT SINGLE QUOTATION MARK	Publish
010	210	13/02	LEFT DOUBLE QUOTATION MARK	Publish
010	211	13/03	RIGHT DOUBLE QUOTATION MARK	Publish
010	212	13/04	PRESCRIPTION, TAKE, RECIPE	Publish
010	214	13/06	MINUTES	Publish
010	215	13/07	SECONDS	Publish
010	217	13/09	LATIN CROSS	Publish
010	218	13/10	HEXAGRAM	Publish
010	219	13/11	FILLED RECTANGLE BULLET	Publish
010	220	13/12	FILLED LEFT TRIANGLE BULLET	Publish
010	221	13/13	FILLED RIGHT TRIANGLE BULLET	Publish
010	222	13/14	EM FILLED CIRCLE	Publish
010	223	13/15	EM FILLED RECTANGLE	Publish
010	224	14/00	EN OPEN CIRCLE BULLET	Publish
010	225	14/01	EN OPEN SQUARE BULLET	Publish
010	226	14/02	OPEN RECTANGULAR BULLET	Publish
010	227	14/03	OPEN TRIANGULAR BULLET UP	Publish
010	228	14/04	OPEN TRIANGULAR BULLET DOWN	Publish
010	229	14/05	OPEN STAR	Publish
010	230	14/06	EN FILLED CIRCLE BULLET	Publish
010	231	14/07	EN FILLED SQUARE BULLET	Publish
010	232	14/08	FILLED TRIANGULAR BULLET UP	Publish
010	233	14/09	FILLED TRIANGULAR BULLET DOWN	Publish
010	234	14/10	LEFT POINTER	Publish
010	235	14/11	RIGHT POINTER	Publish
010	236	14/12	CLUB	Publish
010	237	14/13	DIAMOND	Publish
010	238	14/14	HEART	Publish
010	240	15/00	MALTESE CROSS	Publish
010	241	15/01	DAGGER	Publish
010	242	15/02	DOUBLE DAGGER	Publish
010	243	15/03	CHECK MARK, TICK	Publish
010	244	15/04	BALLOT CROSS	Publish
010	245	15/05	MUSICAL SHARP	Publish
010	246	15/06	MUSICAL FLAT	Publish

Byte 3	Byte 4	Code Pos	Name	Set
010	247	15/07	MALE SYMBOL	Publish
010	248	15/08	FEMALE SYMBOL	Publish
010	249	15/09	TELEPHONE SYMBOL	Publish
010	250	15/10	TELEPHONE RECORDER SYMBOL	Publish
010	251	15/11	PHONOGRAPH COPYRIGHT SIGN	Publish
010	252	15/12	CARET	Publish
010	253	15/13	SINGLE LOW QUOTATION MARK	Publish
010	254	15/14	DOUBLE LOW QUOTATION MARK	Publish
010	255	15/15	CURSOR	Publish
011	163	10/03	LEFT CARET	APL
011	166	10/06	RIGHT CARET	APL
011	168	10/08	DOWN CARET	APL
011	169	10/09	UP CARET	APL
011	192	12/00	OVERBAR	APL
011	194	12/02	DOWN TACK	APL
011	195	12/03	UP SHOE (CAP)	APL
011	196	12/04	DOWN STILE	APL
011	198	12/06	UNDERBAR	APL
011	202	12/10	JOT	APL
011	204	12/12	QUAD	APL
011	206	12/14	UP TACK	APL
011	207	12/15	CIRCLE	APL
011	211	13/03	UP STILE	APL
011	214	13/06	DOWN SHOE (CUP)	APL
011	216	13/08	RIGHT SHOE	APL
011	218	13/10	LEFT SHOE	APL
011	220	13/12	LEFT TACK	APL
011	252	15/12	RIGHT TACK	APL
012	223	13/15	DOUBLE LOW LINE	Hebrew
012	224	14/00	HEBREW LETTER ALEPH	Hebrew
012	225	14/01	HEBREW LETTER BET	Hebrew
012	226	14/02	HEBREW LETTER GIMEL	Hebrew
012	227	14/03	HEBREW LETTER DALET	Hebrew
012	228	14/04	HEBREW LETTER HE	Hebrew
012	229	14/05	HEBREW LETTER WAW	Hebrew
012	230	14/06	HEBREW LETTER ZAIN	Hebrew
012	231	14/07	HEBREW LETTER CHET	Hebrew
012	232	14/08	HEBREW LETTER TET	Hebrew
012	233	14/09	HEBREW LETTER YOD	Hebrew
012	234	14/10	HEBREW LETTER FINAL KAPH	Hebrew
012	235	14/11	HEBREW LETTER KAPH	Hebrew
012	236	14/12	HEBREW LETTER LAMED	Hebrew
012	237	14/13	HEBREW LETTER FINAL MEM	Hebrew
012	238	14/14	HEBREW LETTER MEM	Hebrew
012	239	14/15	HEBREW LETTER FINAL NUN	Hebrew
012	240	15/00	HEBREW LETTER NUN	Hebrew
012	241	15/01	HEBREW LETTER SAMECH	Hebrew
012	242	15/02	HEBREW LETTER A’YIN	Hebrew
012	243	15/03	HEBREW LETTER FINAL PE	Hebrew
012	244	15/04	HEBREW LETTER PE	Hebrew
012	245	15/05	HEBREW LETTER FINAL ZADE	Hebrew
012	246	15/06	HEBREW LETTER ZADE	Hebrew
012	247	15/07	HEBREW QOPH	Hebrew
012	248	15/08	HEBREW RESH	Hebrew
012	249	15/09	HEBREW SHIN	Hebrew
012	250	15/10	HEBREW TAW	Hebrew

Byte 3	Byte 4	Code Pos	Name	Set
255	008	00/08	BACKSPACE, BACK SPACE, BACK CHAR	Keyboard
255	009	00/09	TAB	Keyboard
255	010	00/10	LINEFEED, LF	Keyboard
255	011	00/11	CLEAR	Keyboard
255	013	00/13	RETURN, ENTER	Keyboard
255	019	01/03	PAUSE, HOLD	Keyboard
255	020	01/04	SCROLL LOCK	Keyboard
255	027	01/11	ESCAPE	Keyboard
255	032	02/00	MULTI-KEY CHARACTER PREFACE	Keyboard
255	033	02/01	KANJI, KANJI CONVERT	Keyboard
255	034	02/02	MUHENKAN	Keyboard
255	035	02/03	HENKAN MODE	Keyboard
255	036	02/04	ROMAJI	Keyboard
255	037	02/05	HIRAGANA	Keyboard
255	038	02/06	KATAKANA	Keyboard
255	039	02/07	HIRAGANA/KATAKANA TOGGLE	Keyboard
255	040	02/08	ZENKAKU	Keyboard
255	041	02/09	HANKAKU	Keyboard
255	042	02/10	ZENKAKU/HANKAKU TOGGLE	Keyboard
255	043	02/11	TOUROKU	Keyboard
255	044	02/12	MASSYO	Keyboard
255	045	02/13	KANA LOCK	Keyboard
255	046	02/14	KANA SHIFT	Keyboard
255	047	02/15	EISU SHIFT	Keyboard
255	048	03/00	EISU TOGGLE	Keyboard
255	080	05/00	HOME	Keyboard
255	081	05/01	LEFT, MOVE LEFT, LEFT ARROW	Keyboard
255	082	05/02	UP, MOVE UP, UP ARROW	Keyboard
255	083	05/03	RIGHT, MOVE RIGHT, RIGHT ARROW	Keyboard
255	084	05/04	DOWN, MOVE DOWN, DOWN ARROW	Keyboard
255	085	05/05	PRIOR, PREVIOUS	Keyboard
255	086	05/06	NEXT	Keyboard
255	087	05/07	END, EOL	Keyboard
255	088	05/08	BEGIN, BOL	Keyboard
255	096	06/00	SELECT, MARK	Keyboard
255	097	06/01	PRINT	Keyboard
255	098	06/02	EXECUTE, RUN, DO	Keyboard
255	099	06/03	INSERT, INSERT HERE	Keyboard
255	101	06/05	UNDO, OOPS	Keyboard
255	102	06/06	REDO, AGAIN	Keyboard
255	103	06/07	MENU	Keyboard
255	104	06/08	FIND, SEARCH	Keyboard
255	105	06/09	CANCEL, STOP, ABORT, EXIT	Keyboard
255	106	06/10	HELP, QUESTION MARK	Keyboard
255	107	06/11	BREAK	Keyboard
255	126	07/14	MODE SWITCH, SCRIPT SWITCH, CHARACTER SET SWITCH	Keyboard
255	127	07/15	NUM LOCK	Keyboard
255	128	08/00	KEYPAD SPACE	Keyboard
255	137	08/09	KEYPAD TAB	Keyboard
255	141	08/13	KEYPAD ENTER	Keyboard
255	145	09/01	KEYPAD F1, PF1, A	Keyboard
255	146	09/02	KEYPAD F2, PF2, B	Keyboard
255	147	09/03	KEYPAD F3, PF3, C	Keyboard
255	148	09/04	KEYPAD F4, PF4, D	Keyboard
255	170	10/10	KEYPAD MULTIPLICATION SIGN, ASTERISK	Keyboard
255	171	10/11	KEYPAD PLUS SIGN	Keyboard
255	172	10/12	KEYPAD SEPARATOR, COMMA	Keyboard
255	173	10/13	KEYPAD MINUS SIGN, HYPHEN	Keyboard
255	174	10/14	KEYPAD DECIMAL POINT, FULL STOP	Keyboard
255	175	10/15	KEYPAD DIVISION SIGN, SOLIDUS	Keyboard
255	176	11/00	KEYPAD DIGIT ZERO	Keyboard

Byte 3	Byte 4	Code Pos	Name	Set
255	177	11/01	KEYPAD DIGIT ONE	Keyboard
255	178	11/02	KEYPAD DIGIT TWO	Keyboard
255	179	11/03	KEYPAD DIGIT THREE	Keyboard
255	180	11/04	KEYPAD DIGIT FOUR	Keyboard
255	181	11/05	KEYPAD DIGIT FIVE	Keyboard
255	182	11/06	KEYPAD DIGIT SIX	Keyboard
255	183	11/07	KEYPAD DIGIT SEVEN	Keyboard
255	184	11/08	KEYPAD DIGIT EIGHT	Keyboard
255	185	11/09	KEYPAD DIGIT NINE	Keyboard
255	189	11/13	KEYPAD EQUALS SIGN	Keyboard
255	190	11/14	F1	Keyboard
255	191	11/15	F2	Keyboard
255	192	12/00	F3	Keyboard
255	193	12/01	F4	Keyboard
255	194	12/02	F5	Keyboard
255	195	12/03	F6	Keyboard
255	196	12/04	F7	Keyboard
255	197	12/05	F8	Keyboard
255	198	12/06	F9	Keyboard
255	199	12/07	F10	Keyboard
255	200	12/08	F11, L1	Keyboard
255	201	12/09	F12, L2	Keyboard
255	202	12/10	F13, L3	Keyboard
255	203	12/11	F14, L4	Keyboard
255	204	12/12	F15, L5	Keyboard
255	205	12/13	F16, L6	Keyboard
255	206	12/14	F17, L7	Keyboard
255	207	12/15	F18, L8	Keyboard
255	208	13/00	F19, L9	Keyboard
255	209	13/01	F20, L10	Keyboard
255	210	13/02	F21, R1	Keyboard
255	211	13/03	F22, R2	Keyboard
255	212	13/04	F23, R3	Keyboard
255	213	13/05	F24, R4	Keyboard
255	214	13/06	F25, R5	Keyboard
255	215	13/07	F26, R6	Keyboard
255	216	13/08	F27, R7	Keyboard
255	217	13/09	F28, R8	Keyboard
255	218	13/10	F29, R9	Keyboard
255	219	13/11	F30, R10	Keyboard
255	220	13/12	F31, R11	Keyboard
255	221	13/13	F32, R12	Keyboard
255	222	13/14	F33, R13	Keyboard
255	223	13/15	F34, R14	Keyboard
255	224	14/00	F35, R15	Keyboard
255	225	14/01	LEFT SHIFT	Keyboard
255	226	14/02	RIGHT SHIFT	Keyboard
255	227	14/03	LEFT CONTROL	Keyboard
255	228	14/04	RIGHT CONTROL	Keyboard
255	229	14/05	CAPS LOCK	Keyboard
255	230	14/06	SHIFT LOCK	Keyboard
255	231	14/07	LEFT META	Keyboard
255	232	14/08	RIGHT META	Keyboard
255	233	14/09	LEFT ALT	Keyboard
255	234	14/10	RIGHT ALT	Keyboard
255	235	14/11	LEFT SUPER	Keyboard
255	236	14/12	RIGHT SUPER	Keyboard
255	237	14/13	LEFT HYPER	Keyboard
255	238	14/14	RIGHT HYPER	Keyboard
255	255	15/15	DELETE, RUBOUT	Keyboard

Appendix B

Protocol Encoding

Syntactic Conventions

All numbers are in decimal, unless prefixed with #x, in which case they are in hexadecimal (base 16).

The general syntax used to describe requests, replies, errors, events, and compound types is:

NameofThing
encode-form
...
encode-form

Each encode-form describes a single component.

For components described in the protocol as:

name: TYPE

the encode-form is:

N	TYPE	name

N is the number of bytes occupied in the data stream, and TYPE is the interpretation of those bytes. For example,

depth: CARD8

becomes:

1	CARD8	depth

For components with a static numeric value the encode-form is:

N	value	name

The value is always interpreted as an N-byte unsigned integer. For example, the first two bytes of a Window error are always zero (indicating an error in general) and three (indicating the Window error in particular):

1	0	Error
1	3	code

For components described in the protocol as:

name: {**Name1**,..., **NameI**}

the encode-form is:

N		name
	value1 Name1	
	...	
	valueI NameI	

The value is always interpreted as an N-byte unsigned integer. Note that the size of N is sometimes larger than that strictly required to encode the values. For example:

class: {**InputOutput**, **InputOnly**, **CopyFromParent**}

becomes:

2			class
	0	CopyFromParent	
	1	InputOutput	
	2	InputOnly	

For components described in the protocol as:

NAME: TYPE or **Alternative1**...or **AlternativeI** the encode-form is:

N	TYPE		NAME
	value1	Alternative1	
	...		
	valueI	AlternativeI	

The alternative values are guaranteed not to conflict with the encoding of TYPE. For example:

destination: WINDOW or **PointerWindow** or **InputFocus**

becomes:

4	WINDOW		destination
	0	PointerWindow	
	1	InputFocus	

For components described in the protocol as:

value-mask: BITMASK

the encode-form is:

N	BITMASK		value-mask
	mask1	mask-name1	
	...		
	maskI	mask-nameI	

The individual bits in the mask are specified and named, and N is 2 or 4. The most-significant bit in a BITMASK is reserved for use in defining chained (multiword) bitmasks, as extensions augment existing core requests. The precise interpretation of this bit is not yet defined here, although a probable mechanism is that a 1-bit indicates that another N bytes of bitmask follows, with bits within the overall mask still interpreted from least-significant to most-significant with an N-byte unit, with N-byte units interpreted in stream order, and with the overall mask being byte-swapped in individual N-byte units.

For LISTofVALUE encodings, the request is followed by a section of the form:

VALUEs
 encode-form
 ...
 encode-form

listing an encode-form for each VALUE. The NAME in each encode-form keys to the corresponding BITMASK bit. The encoding of a VALUE always occupies four bytes, but the number of bytes specified in the encoding-form indicates how many of the least-significant bytes are actually used; the remaining bytes are unused and their values do not matter.

In various cases, the number of bytes occupied by a component will be specified by a lowercase single-letter variable name instead of a specific numeric value, and often some other component will have its value specified as a simple numeric expression involving these variables. Components specified with such expressions are always interpreted as unsigned integers. The scope of such variables is always just the enclosing request, reply, error, event, or compound type structure. For example:

2	3+n	request length
4n	LISTofPOINT	points

For unused bytes (the values of the bytes are undefined and do no matter), the encode-form is:

N	unused

If the number of unused bytes is variable, the encode-form typically is:

p	unused, p=pad(E)

where E is some expression, and pad(E) is the number of bytes needed to round E up to a multiple of four.

pad(E) = (4 - (E mod 4)) mod 4

Common Types

LISTofFOO

In this document the LISTof notation strictly means some number of repetitions of the FOO encoding; the actual length of the list is encoded elsewhere.

SETofFOO

A set is always represented by a bitmask, with a 1-bit indicating presence in the set.

BITMASK: CARD32

WINDOW: CARD32

PIXMAP: CARD32

CURSOR: CARD32

FONT: CARD32

GCONTEXT: CARD32

COLORMAP: CARD32

DRAWABLE: CARD32

FONTABLE: CARD32

ATOM: CARD32

VISUALID: CARD32

BYTE: 8-bit value

INT8: 8-bit signed integer

INT16: 16-bit signed integer

INT32: 32-bit signed integer

CARD8: 8-bit unsigned integer

CARD16: 16-bit unsigned integer

CARD32: 32-bit unsigned integer

TIMESTAMP: CARD32

BITGRAVITY

0	Forget
1	NorthWest
2	North
3	NorthEast
4	West
5	Center
6	East
7	SouthWest
8	South
9	SouthEast
10	Static

WINGRAVITY

0	Unmap
1	NorthWest
2	North
3	NorthEast
4	West
5	Center
6	East
7	SouthWest
8	South
9	SouthEast
10	Static

BOOL

0	False
1	True

SETofEVENT

#x00000001	KeyPress

#x00000002	KeyRelease
#x00000004	ButtonPress
#x00000008	ButtonRelease
#x00000010	EnterWindow
#x00000020	LeaveWindow
#x00000040	PointerMotion
#x00000080	PointerMotionHint
#x00000100	Button1Motion
#x00000200	Button2Motion
#x00000400	Button3Motion
#x00000800	Button4Motion
#x00001000	Button5Motion
#x00002000	ButtonMotion
#x00004000	KeymapState
#x00008000	Exposure
#x00010000	VisibilityChange
#x00020000	StructureNotify
#x00040000	ResizeRedirect
#x00080000	SubstructureNotify
#x00100000	SubstructureRedirect
#x00200000	FocusChange
#x00400000	PropertyChange
#x00800000	ColormapChange
#x01000000	OwnerGrabButton
#xfe000000	unused but must be zero

SETofPOINTEREVENT

encodings are the same as for SETofEVENT, except with

#xffff8003	unused but must be zero

SETofDEVICEEVENT

encodings are the same as for SETofEVENT, except with

#xffffc0b0	unused but must be zero

KEYSYM: CARD32

KEYCODE: CARD8

BUTTON: CARD8

SETofKEYBUTMASK

#x0001	Shift
#x0002	Lock
#x0004	Control
#x0008	Mod1
#x0010	Mod2
#x0020	Mod3
#x0040	Mod4
#x0080	Mod5
#x0100	Button1
#x0200	Button2
#x0400	Button3
#x0800	Button4
#x1000	Button5
#xe000	unused but must be zero

SETofKEYMASK

encodings are the same as for SETofKEYBUTMASK, except with

#xff00	unused but must be zero

STRING8: LISTofCARD8

STRING16: LISTofCHAR2B

CHAR2B

1	CARD8	byte1
1	CARD8	byte2

POINT

2	INT16	x	
2	INT16	y	

RECTANGLE

2	INT16	x	
2	INT16	y	
2	CARD16	width	
2	CARD16	height	

ARC

2	INT16	x	
2	INT16	y	
2	CARD16	width	
2	CARD16	height	
2	INT16	angle1	
2	INT16	angle2	

HOST

1			family
	0	Internet	
	1	DECnet	
	2	Chaos	
1			unused
2	n		length of address
n	LISTofBYTE		address
p			unused, p=pad(n)

STR

1	n		length of name in bytes
n	STRING8		name

Errors

Request

1	0		Error
1	1		code
2	CARD16		sequence number
4			unused
2	CARD16		minor opcode
1	CARD8		major opcode
21			unused

Value

1	0		Error
1	2		code
2	CARD16		sequence number
4	<32-bits>		bad value
2	CARD16		minor opcode
1	CARD8		major opcode
21			unused

Window

1	0		Error
1	3		code
2	CARD16		sequence number
4	CARD32		bad resource id
2	CARD16		minor opcode
1	CARD8		major opcode
21			unused

Pixmap

1	0		Error
1	4		code
2	CARD16		sequence number
4	CARD32		bad resource id

2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Atom

1	0	Error
1	5	code
2	CARD16	sequence number
4	CARD32	bad atom id
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Cursor

1	0	Error
1	6	code
2	CARD16	sequence number
4	CARD32	bad resource id
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Font

1	0	Error
1	7	code
2	CARD16	sequence number
4	CARD32	bad resource id
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Match

1	0	Error
1	8	code
2	CARD16	sequence number
4		unused
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Drawable

1	0	Error
1	9	code
2	CARD16	sequence number
4	CARD32	bad resource id
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Access

1	0	Error
1	10	code
2	CARD16	sequence number
4		unused
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Alloc

1	0	Error
1	11	code
2	CARD16	sequence number
4		unused
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Colormap

1	0	Error
1	12	code
2	CARD16	sequence number
4	CARD32	bad resource id
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

GContext

1	0	Error
1	13	code
2	CARD16	sequence number
4	CARD32	bad resource id
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

IDChoice

1	0	Error
1	14	code
2	CARD16	sequence number
4	CARD32	bad resource id
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Name

1	0	Error
1	15	code
2	CARD16	sequence number
4		unused
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Length

1	0	Error
1	16	code
2	CARD16	sequence number
4		unused
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Implementation

1	0	Error
1	17	code
2	CARD16	sequence number
4		unused
2	CARD16	minor opcode
1	CARD8	major opcode
21		unused

Keyboards

KEYCODE values are always greater than 7 (and less than 256).

KEYSYM values with the bit #x10000000 set are reserved as vendor-specific.

The names and encodings of the standard KEYSYM values are contained in appendix F.

Pointers

BUTTON values are numbered starting with one.

Predefined Atoms

PRIMARY	1
SECONDARY	2
ARC	3
ATOM	4
BITMAP	5
CARDINAL	6
COLORMAP	7
CURSOR	8
CUT_BUFFER0	9
CUT_BUFFER1	10
CUT_BUFFER2	11
CUT_BUFFER3	12
CUT_BUFFER4	13
CUT_BUFFER5	14
CUT_BUFFER6	15
CUT_BUFFER7	16
DRAWABLE	17
FONT	18
INTEGER	19
PIXMAP	20
POINT	21
RECTANGLE	22
RESOURCE_MANGER	23
RGB_COLOR_MAP	24
RGB_BEST_MAP	25
RGB_BLUE_MAP	26
RGB_DEFAULT_MAP	27
RGB_GRAY_MAP	28
RGB_GREEN_MAP	29
RGB_RED_MAP	30
STRING	31
VISUALID	32
WINDOW	33
WM_COMMAND	34
WM_HINTS	35
WM_CLIENT_MACHINE	36
WM_ICON_NAME	37
WM_ICON_SIZE	38
WM_NAME	39
WM_NORMAL_HINTS	40
WM_SIZE_HINTS	41
WM_ZOOM_HINTS	42
MIN_SPACE	43
NORM_SPACE	44
MAX_SPACE	45
END_SPACE	46
SUPERSCRIPT_X	47
SUPERSCRIPT_Y	48
SUBSCRIPT_X	49
SUBSCRIPT_Y	50
UNDERLINE_POSITION	51
UNDERLINE_THICKNESS	52
STRIKEOUT_ASCENT	53
STRIKEOUT_DESCENT	54
ITALIC_ANGLE	55
X_HEIGHT	56
QUAD_WIDTH	57
WEIGHT	58
POINT_SIZE	59
RESOLUTION	60
COPYRIGHT	61
NOTICE	62
FONT_NAME	63
FAMILY_NAME	64
FULL_NAME	65
CAP_HEIGHT	66
WM_CLASS	67
WM_TRANSIENT_FOR	68

Connection Setup

For TCP connections, displays on a given host are numbered starting from 0, and the server for display N listens and accepts connections on port 6000 + N. For DECnet connections, displays on a given host are numbered starting from 0, and the server for display N listens and accepts connections on the object name obtained by concatenating ‘‘X$X’’ with the decimal representation of N, for example, X$X0 and X$X1.

Information sent by the client at connection setup:

1			byte-order
	#x42	MSB first	
	#x6C	LSB first	
1			unused
2	CARD16		protocol-major-version
2	CARD16		protocol-minor-version
2	n		length of authorization-protocol-name
2	d		length of authorization-protocol-data
2			unused
n	STRING8		authorization-protocol-name
p			unused, p=pad(n)
d	STRING8		authorization-protocol-data
q			unused, q=pad(d)

Except where explicitly noted in the protocol, all 16-bit and 32-bit quantities sent by the client must be transmitted with the specified byte order, and all 16-bit and 32-bit quantities returned by the server will be transmitted with this byte order.

Information received by the client if authorization fails:

1	0		failed
1	n		length of reason in bytes
2	CARD16		protocol-major-version
2	CARD16		protocol-minor-version
2	(n+p)/4		length in 4-byte units of ‘‘additional data’’
n	STRING8		reason
p			unused, p=pad(n)

Information received by the client if authorization is accepted:

1	1		success
1			unused
2	CARD16		protocol-major-version
2	CARD16		protocol-minor-version
2	8+2n+(v+p+m)/4		length in 4-byte units of ‘‘additional data’’
4	CARD32		release-number
4	CARD32		resource-id-base
4	CARD32		resource-id-mask
4	CARD32		motion-buffer-size
2	v		length of vendor
2	CARD16		maximum-request-length
1	CARD8		number of SCREENs in roots
1	n		number for FORMATs in pixmap-formats
1			image-byte-order
	0	LSBFirst	
	1	MSBFirst	
1			bitmap-format-bit-order
	0	LeastSignificant	
	1	MostSignificant	
1	CARD8		bitmap-format-scanline-unit
1	CARD8		bitmap-format-scanline-pad
1	KEYCODE		min-keycode
1	KEYCODE		max-keycode
4			unused
v	STRING8		vendor
p			unused, p=pad(v)
8n	LISTofFORMAT		pixmap-formats
m	LISTofSCREEN		roots (m is always a multiple of 4)

FORMAT

1	CARD8		depth
1	CARD8		bits-per-pixel
1	CARD8		scanline-pad
5			unused

SCREEN

4	WINDOW		root
4	COLORMAP		default-colormap
4	CARD32		white-pixel
4	CARD32		black-pixel
4	SETofEVENT		current-input-masks
2	CARD16		width-in-pixels
2	CARD16		height-in-pixels
2	CARD16		width-in-millimeters
2	CARD16		height-in-millimeters
2	CARD16		min-installed-maps
2	CARD16		max-installed-maps
4	VISUALID		root-visual
1			backing-stores
	0	Never	
	1	WhenMapped	
	2	Always	
1	BOOL		save-unders
1	CARD8		root-depth
1	CARD8		number of DEPTHs in allowed-depths

n	LISTofDEPTH		allowed-depths (n is always a multiple of 4)

DEPTH

1	CARD8		depth
1			unused
2	n		number of VISUALTYPES in visuals
4			unused
24n	LISTofVISUALTYPE		visuals

VISUALTYPE

4	VISUALID		visual-id
1			class
	0	StaticGray	
	1	GrayScale	
	2	StaticColor	
	3	PseudoColor	
	4	TrueColor	
	5	DirectColor	
1	CARD8		bits-per-rgb-value
2	CARD16		colormap-entries
4	CARD32		red-mask
4	CARD32		green-mask
4	CARD32		blue-mask
4			unused

Requests

CreateWindow

1	1		opcode
1	CARD8		depth
2	8+n		request length
4	WINDOW		wid
4	WINDOW		parent
2	INT16		x
2	INT16		y
2	CARD16		width
2	CARD16		height
2	CARD16		border-width
2			class
	0	CopyFromParent	
	1	InputOutput	
	2	InputOnly	
4	VISUALID		visual
	0	CopyFromParent	
4	BITMASK		value-mask (has n bits set to 1)
	#x00000001	background-pixmap	
	#x00000002	background-pixel	
	#x00000004	border-pixmap	
	#x00000008	border-pixel	
	#x00000010	bit-gravity	
	#x00000020	win-gravity	
	#x00000040	backing-store	
	#x00000080	backing-planes	
	#x00000100	backing-pixel	
	#x00000200	override-redirect	
	#x00000400	save-under	
	#x00000800	event-mask	
	#x00001000	do-not-propagate-mask	
	#x00002000	colormap	
	#x00004000	cursor	
4n	LISTofVALUE		value-list

VALUEs

4	PIXMAP		background-pixmap
	0	None	
	1	ParentRelative	

4	CARD32		background-pixel
4	PIXMAP		border-pixmap
	0	CopyFromParent	
4	CARD32		border-pixel
1	BITGRAVITY		bit-gravity
1	WINGRAVITY		win-gravity
1			backing-store
	0	NotUseful	
	1	WhenMapped	
	2	Always	
4	CARD32		backing-planes
4	CARD32		backing-pixel
1	BOOL		override-redirect
1	BOOL		save-under
4	SETofEVENT		event-mask
4	SETofDEVICEEVENT		do-not-propagate-mask
4	COLORMAP		colormap
	0	CopyFromParent	
4	CURSOR		cursor
	0	None	

ChangeWindowAttributes

1	2		opcode
1			unused
2	3+n		request length
4	WINDOW		window
4	BITMASK		value-mask (has n bits set to 1)
	encodings are the same as for CreateWindow		
4n	LISTofVALUE		value-list
	encodings are the same as for CreateWindow		

GetWindowAttributes

1	3		opcode
1			unused
2	2		request length
4	WINDOW		window

=>

1	1		Reply
1			backing-store
	0	NotUseful	
	1	WhenMapped	
	2	Always	
2	CARD16		sequence number
4	3		reply length
4	VISUALID		visual
2			class
	1	InputOutput	
	2	InputOnly	
1	BITGRAVITY		bit-gravity
1	WINGRAVITY		win-gravity
4	CARD32		backing-planes
4	CARD32		backing-pixel
1	BOOL		save-under
1	BOOL		map-is-installed
1			map-state
	0	Unmapped	
	1	Unviewable	
	2	Viewable	
1	BOOL		override-redirect
4	COLORMAP		colormap
	0	None	
4	SETofEVENT		all-event-masks
4	SETofEVENT		your-event-mask
2	SETofDEVICEEVENT		do-not-propagate-mask
2			unused

DestroyWindow

1	4		opcode
1			unused
2	2		request length
4	WINDOW		window

DestroySubwindows

1	5		opcode
1			unused
2	2		request length
4	WINDOW		window

ChangeSaveSet

1	6		opcode
1			mode
	0	Insert	
	1	Delete	
2	2		request length
4	WINDOW		window

ReparentWindow

1	7		opcode
1			unused
2	4		request length
4	WINDOW		window
4	WINDOW		parent
2	INT16		x
2	INT16		y

MapWindow

1	8		opcode
1			unused
2	2		request length
4	WINDOW		window

MapSubwindows

1	9		opcode
1			unused
2	2		request length
4	WINDOW		window

UnmapWindow

1	10		opcode
1			unused
2	2		request length
4	WINDOW		window

UnmapSubwindows

1	11		opcode
1			unused
2	2		request length
4	WINDOW		window

ConfigureWindow

1	12		opcode
1			unused
2	3+n		request length
4	WINDOW		window
2	BITMASK		value-mask (has n bits set to 1)
	#x0001	x	
	#x0002	y	
	#x0004	width	
	#x0008	height	
	#x0010	border-width	
	#x0020	sibling	
	#x0040	stack-mode	

2		unused	
4n	LISTofVALUE		value-list

VALUEs

2	INT16		x
2	INT16		y
2	CARD16		width
2	CARD16		height
2	CARD16		border-width
4	WINDOW		sibling
1			stack-mode
	0	Above	
	1	Below	
	2	TopIf	
	3	BottomIf	
	4	Opposite	

CirculateWindow

1	13		opcode
1			direction
	0	RaiseLowest	
	1	LowerHighest	
2	2		request length
4	WINDOW		window

GetGeometry

1	14		opcode
1			unused
2	2		request length
4	DRAWABLE		drawable

=>

1	1		Reply
1	CARD8		depth
2	CARD16		sequence number
4	0		reply length
4	WINDOW		root
2	INT16		x
2	INT16		y
2	CARD16		width
2	CARD16		height
2	CARD16		border-width
10			unused

QueryTree

1	15		opcode
1			unused
2	2		request length
4	WINDOW		window

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	n		reply length
4	WINDOW		root
4	WINDOW		parent
	0	None	
2	n		number of WINDOWs in children
14			unused
4n	LISTofWINDOW		children

InternAtom

1	16		opcode
1	BOOL		only-if-exists
2	2+(n+p)/4		request length
2	n		length of name

2		unused	
n	STRING8		name
p			unused, p=pad(n)

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	0		reply length
4	ATOM		atom
	0	None	
20			unused

GetAtomName

1	17		opcode
1			unused
2	2		request length
4	ATOM		atom

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	(n+p)/4		reply length
2	n		length of name
22			unused
n	STRING8		name
p			unused, p=pad(n)

ChangeProperty

1	18		opcode
1			mode
	0	Replace	
	1	Prepend	
	2	Append	
2	6+(n+p)/4		request length
4	WINDOW		window
4	ATOM		property
4	ATOM		type
1	CARD8		format
3			unused
4	CARD32		length of data in format units (= n for format = 8) (= n/2 for format = 16) (= n/4 for format = 32)
n	LISTofBYTE		data (n is a multiple of 2 for format = 16) (n is a multiple of 4 for format = 32)
p			unused, p=pad(n)

DeleteProperty

1	19		opcode
1			unused
2	3		request length
4	WINDOW		window
4	ATOM		property

GetProperty

1	20		opcode
1	BOOL		delete
2	6		request length
4	WINDOW		window
4	ATOM		property
4	ATOM		type
	0	AnyPropertyType	
4	CARD32		long-offset

4	CARD32		long-length

=>

1	1		Reply
1	CARD8		format
2	CARD16		sequence number
4	(n+p)/4		reply length
4	ATOM		type
	0	None	
4	CARD32		bytes-after
4	CARD32		length of value in format units (= 0 for format = 0) (= n for format = 8) (= n/2 for format = 16) (= n/4 for format = 32)
12			unused
n	LISTofBYTE		value (n is zero for format = 0) (n is a multiple of 2 for format = 16) (n is a multiple of 4 for format = 32)
p			unused, p=pad(n)

ListProperties

1	21		opcode
1			unused
2	2		request length
4	WINDOW		window

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	n		reply length
2	n		number of ATOMs in atoms
22			unused
4n	LISTofATOM		atoms

SetSelectionOwner

1	22		opcode
1			unused
2	4		request length
4	WINDOW		owner
	0	None	
4	ATOM		selection
4	TIMESTAMP		time
	0	CurrentTime	

GetSelectionOwner

1	23		opcode
1			unused
2	2		request length
4	ATOM		selection

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	0		reply length
4	WINDOW		owner
	0	None	
20			unused

ConvertSelection

1	24		opcode
1			unused
2	6		request length
4	WINDOW		requestor

4	ATOM		selection
4	ATOM		target
4	ATOM		property
	0	None	
4	TIMESTAMP		time
	0	CurrentTime	

SendEvent

1	25		opcode
1	BOOL		propagate
2	11		request length
4	WINDOW		destination
	0	PointerWindow	
	1	InputFocus	
4	SETofEVENT		event-mask
32			event
	standard event format (see the Events section)		

GrabPointer

1	26		opcode
1	BOOL		owner-events
2	6		request length
4	WINDOW		grab-window
2	SETofPOINTEREVENT		event-mask
1			pointer-mode
	0	Synchronous	
	1	Asynchronous	
1			keyboard-mode
	0	Synchronous	
	1	Asynchronous	
4	WINDOW		confine-to
	0	None	
4	CURSOR		cursor
	0	None	
4	TIMESTAMP		time
	0	CurrentTime	

=>

1	1		Reply
1			status
	0	Success	
	1	AlreadyGrabbed	
	2	InvalidTime	
	3	NotViewable	
	4	Frozen	
2	CARD16		sequence number
4	0		reply length
24			unused

UngrabPointer

1	27		opcode
1			unused
2	2		request length
4	TIMESTAMP		time
	0	CurrentTime	

GrabButton

1	28		opcode
1	BOOL		owner-events
2	6		request length
4	WINDOW		grab-window
2	SETofPOINTEREVENT		event-mask
1			pointer-mode
	0	Synchronous	
	1	Asynchronous	
1			keyboard-mode

0	Synchronous		
	1	Asynchronous	
4	WINDOW		confine-to
	0	None	
4	CURSOR		cursor
	0	None	
1	BUTTON		button
	0	AnyButton	
1			unused
2	SETofKEYMASK		modifiers
	#x8000	AnyModifier	

UngrabButton

1	29		opcode
1	BUTTON		button
	0	AnyButton	
2	3		request length
4	WINDOW		grab-window
2	SETofKEYMASK		modifiers
	#x8000	AnyModifier	
2			unused

ChangeActivePointerGrab

1	30		opcode
1			unused
2	4		request length
4	CURSOR		cursor
	0	None	
4	TIMESTAMP		time
	0	CurrentTime	
2	SETofPOINTEREVENT		event-mask
2			unused

GrabKeyboard

1	31		opcode
1	BOOL		owner-events
2	4		request length
4	WINDOW		grab-window
4	TIMESTAMP		time
	0	CurrentTime	
1			pointer-mode
	0	Synchronous	
	1	Asynchronous	
1			keyboard-mode
	0	Synchronous	
	1	Asynchronous	
2			unused

=>

1	1		Reply
1			status
	0	Success	
	1	AlreadyGrabbed	
	2	InvalidTime	
	3	NotViewable	
	4	Frozen	
2	CARD16		sequence number
4	0		reply length
24			unused

UngrabKeyboard

1	32		opcode
1			unused
2	2		request length
4	TIMESTAMP		time
	0	CurrentTime	

GrabKey

1	33		opcode
1	BOOL		owner-events
2	4		request length
4	WINDOW		grab-window
2	SETofKEYMASK		modifiers
	#x8000	AnyModifier	
1	KEYCODE		key
	0	AnyKey	
1			pointer-mode
	0	Synchronous	
	1	Asynchronous	
1			keyboard-mode
	0	Synchronous	
	1	Asynchronous	
3			unused

UngrabKey

1	34		opcode
1	KEYCODE		key
	0	AnyKey	
2	3		request length
4	WINDOW		grab-window
2	SETofKEYMASK		modifiers
	#x8000	AnyModifier	
2			unused

AllowEvents

1	35		opcode
1			mode
	0	AsyncPointer	
	1	SyncPointer	
	2	ReplayPointer	
	3	AsyncKeyboard	
	4	SyncKeyboard	
	5	ReplayKeyboard	
	6	AsyncBoth	
	7	SyncBoth	
2	2		request length
4	TIMESTAMP		time
	0	CurrentTime	

GrabServer

1	36		opcode
1			unused
2	1		request length

UngrabServer

1	37		opcode
1			unused
2	1		request length

QueryPointer

1	38		opcode
1			unused
2	2		request length
4	WINDOW		window

=>

1	1		Reply
1	BOOL		same-screen
2	CARD16		sequence number
4	0		reply length
4	WINDOW		root
4	WINDOW		child
	0	None	

2	INT16		root-x
2	INT16		root-y
2	INT16		win-x
2	INT16		win-y
2	SETofKEYBUTMASK		mask
6			unused

GetMotionEvents

1	39		opcode
1			unused
2	4		request length
4	WINDOW		window
4	TIMESTAMP		start
	0	CurrentTime	
4	TIMESTAMP		stop
	0	CurrentTime	

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	2n		reply length
4	n		number of TIMECOORDs in events
20			unused
8n	LISTofTIMECOORD		events

TIMECOORD

4	TIMESTAMP		time
2	CARD16		x
2	CARD16		y

TranslateCoordinates

1	40		opcode
1			unused
2	4		request length
4	WINDOW		src-window
4	WINDOW		dst-window
2	INT16		src-x
2	INT16		src-y

=>

1	1		Reply
1	BOOL		same-screen
2	CARD16		sequence number
4	0		reply length
4	WINDOW		child
	0	None	
2	INT16		dst-x
2	INT16		dst-y
16			unused

WarpPointer

1	41		opcode
1			unused
2	6		request length
4	WINDOW		src-window
	0	None	
4	WINDOW		dst-window
	0	None	
2	INT16		src-x
2	INT16		src-y
2	CARD16		src-width
2	CARD16		src-height
2	INT16		dst-x
2	INT16		dst-y

SetInputFocus

1	42		opcode
1			revert-to
	0	None	
	1	PointerRoot	
	2	Parent	
2	3		request length
4	WINDOW		focus
	0	None	
	1	PointerRoot	
4	TIMESTAMP		time
	0	CurrentTime	

GetInputFocus

1	43		opcode
1			unused
2	1		request length

=>

1	1		Reply
1			revert-to
	0	None	
	1	PointerRoot	
	2	Parent	
2	CARD16		sequence number
4	0		reply length
4	WINDOW		focus
	0	None	
	1	PointerRoot	
20			unused

QueryKeymap

1	44		opcode
1			unused
2	1		request length

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	2		reply length
32	LISTofCARD8		keys

OpenFont

1	45		opcode
1			unused
2	3+(n+p)/4		request length
4	FONT		fid
2	n		length of name
2			unused
n	STRING8		name
p			unused, p=pad(n)

CloseFont

1	46		opcode
1			unused
2	2		request length
4	FONT		font

QueryFont

1	47		opcode
1			unused
2	2		request length
4	FONTABLE		font

=>

1	1		Reply

1		unused	
2	CARD16		sequence number
4	7+2n+3m		reply length
12	CHARINFO		min-bounds
4			unused
12	CHARINFO		max-bounds
4			unused
2	CARD16		min-char-or-byte2
2	CARD16		max-char-or-byte2
2	CARD16		default-char
2	n		number of FONTPROPs in properties
1			draw-direction
	0	LeftToRight	
	1	RightToLeft	
1	CARD8		min-byte1
1	CARD8		max-byte1
1	BOOL		all-chars-exist
2	INT16		font-ascent
2	INT16		font-descent
4	m		number of CHARINFOs in char-infos
8n	LISTofFONTPROP		properties
12m	LISTofCHARINFO		char-infos

FONTPROP

4	ATOM	name
4	<32-bits>	value

CHARINFO

2	INT16	left-side-bearing
2	INT16	right-side-bearing
2	INT16	character-width
2	INT16	ascent
2	INT16	descent
2	CARD16	attributes

QueryTextExtents

1	48	opcode
1	BOOL	odd length, True if p = 2
2	2+(2n+p)/4	request length
4	FONTABLE	font
2n	STRING16	string
p		unused, p=pad(2n)

=>

1	1		Reply
1			draw-direction
	0	LeftToRight	
	1	RightToLeft	
2	CARD16		sequence number
4	0		reply length
2	INT16		font-ascent
2	INT16		font-descent
2	INT16		overall-ascent
2	INT16		overall-descent
4	INT32		overall-width
4	INT32		overall-left
4	INT32		overall-right
4			unused

ListFonts

1	49	opcode
1		unused
2	2+(n+p)/4	request length
2	CARD16	max-names
2	n	length of pattern
n	STRING8	pattern

p		unused, p=pad(n)	

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	(n+p)/4		reply length
2	CARD16		number of STRs in names
22			unused
n	LISTofSTR		names
p			unused, p=pad(n)

ListFontsWithInfo

1	50		opcode
1			unused
2	2+(n+p)/4		request length
2	CARD16		max-names
2	n		length of pattern
n	STRING8		pattern
p			unused, p=pad(n)

=> (except for last in series)

1	1		Reply
1	n		length of name in bytes
2	CARD16		sequence number
4	7+2m+(n+p)/4		reply length
12	CHARINFO		min-bounds
4			unused
12	CHARINFO		max-bounds
4			unused
2	CARD16		min-char-or-byte2
2	CARD16		max-char-or-byte2
2	CARD16		default-char
2	m		number of FONTPROPs in properties
1			draw-direction
	0	LeftToRight	
	1	RightToLeft	
1	CARD8		min-byte1
1	CARD8		max-byte1
1	BOOL		all-chars-exist
2	INT16		font-ascent
2	INT16		font-descent
4	CARD32		replies-hint
8m	LISTofFONTPROP		properties
n	STRING8		name
p			unused, p=pad(n)

FONTPROP
encodings are the same as for QueryFont

CHARINFO
encodings are the same as for QueryFont

=> (last in series)

1	1		Reply
1	0		last-reply indicator
2	CARD16		sequence number
4	7		reply length
52			unused

SetFontPath

1	51		opcode
1			unused
2	2+(n+p)/4		request length
2	CARD16		number of STRs in path
2			unused
n	LISTofSTR		path

p		unused, p=pad(n)	

GetFontPath

1	52		opcode
1			unused
2	1		request list

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	(n+p)/4		reply length
2	CARD16		number of STRs in path
22			unused
n	LISTofSTR		path
p			unused, p=pad(n)

CreatePixmap

1	53		opcode
1	CARD8		depth
2	4		request length
4	PIXMAP		pid
4	DRAWABLE		drawable
2	CARD16		width
2	CARD16		height

FreePixmap

1	54		opcode
1			unused
2	2		request length
4	PIXMAP		pixmap

CreateGC

1	55		opcode
1			unused
2	4+n		request length
4	GCONTEXT		cid
4	DRAWABLE		drawable
4	BITMASK		value-mask (has n bits set to 1)
	#x00000001	function	
	#x00000002	plane-mask	
	#x00000004	foreground	
	#x00000008	background	
	#x00000010	line-width	
	#x00000020	line-style	
	#x00000040	cap-style	
	#x00000080	join-style	
	#x00000100	fill-style	
	#x00000200	fill-rule	
	#x00000400	tile	
	#x00000800	stipple	
	#x00001000	tile-stipple-x-origin	
	#x00002000	tile-stipple-y-origin	
	#x00004000	font	
	#x00008000	subwindow-mode	
	#x00010000	graphics-exposures	
	#x00020000	clip-x-origin	
	#x00040000	clip-y-origin	
	#x00080000	clip-mask	
	#x00100000	dash-offset	
	#x00200000	dashes	
	#x00400000	arc-mode	
4n	LISTofVALUE		value-list

VALUEs

1			function

0	Clear		
	1	And	
	2	AndReverse	
	3	Copy	
	4	AndInverted	
	5	NoOp	
	6	Xor	
	7	Or	
	8	Nor	
	9	Equiv	
	10	Invert	
	11	OrReverse	
	12	CopyInverted	
	13	OrInverted	
	14	Nand	
	15	Set	
4	CARD32		plane-mask
4	CARD32		foreground
4	CARD32		background
2	CARD16		line-width
1			line-style
	0	Solid	
	1	OnOffDash	
	2	DoubleDash	
1			cap-style
	0	NotLast	
	1	Butt	
	2	Round	
	3	Projecting	
1			join-style
	0	Miter	
	1	Round	
	2	Bevel	
1			fill-style
	0	Solid	
	1	Tiled	
	2	Stippled	
	3	OpaqueStippled	
1			fill-rule
	0	EvenOdd	
	1	Winding	
4	PIXMAP		tile
4	PIXMAP		stipple
2	INT16		tile-stipple-x-origin
2	INT16		tile-stipple-y-origin
4	FONT		font
1			subwindow-mode
	0	ClipByChildren	
	1	IncludeInferiors	
1	BOOL		graphics-exposures
2	INT16		clip-x-origin
2	INT16		clip-y-origin
4	PIXMAP		clip-mask
	0	None	
2	CARD16		dash-offset
1	CARD8		dashes
1			arc-mode
	0	Chord	
	1	PieSlice	

ChangeGC

1	56	opcode
1		unused
2	3+n	request length
4	GCONTEXT	gc

4	BITMASK	value-mask (has n bits set to 1)
	encodings are the same as for CreateGC	
4n	LISTofVALUE	value-list
	encodings are the same as for CreateGC	

CopyGC

1	57	opcode
1		unused
2	4	request length
4	GCONTEXT	src-gc
4	GCONTEXT	dst-gc
4	BITMASK	value-mask
	encodings are the same as for CreateGC	

SetDashes

1	58	opcode
1		unused
2	3+(n+p)/4	request length
4	GCONTEXT	gc
2	CARD16	dash-offset
2	n	length of dashes
n	LISTofCARD8	dashes
p		unused, p=pad(n)

SetClipRectangles

1	59		opcode
1			ordering
	0	UnSorted	
	1	YSorted	
	2	YXSorted	
	3	YXBanded	
2	3+2n		request length
4	GCONTEXT		gc
2	INT16		clip-x-origin
2	INT16		clip-y-origin
8n	LISTofRECTANGLE		rectangles

FreeGC

1	60	opcode
1		unused
2	2	request length
4	GCONTEXT	gc

ClearArea

1	61	opcode
1	BOOL	exposures
2	4	request length
4	WINDOW	window
2	INT16	x
2	INT16	y
2	CARD16	width
2	CARD16	height

CopyArea

1	62	opcode
1		unused
2	7	request length
4	DRAWABLE	src-drawable
4	DRAWABLE	dst-drawable
4	GCONTEXT	gc
2	INT16	src-x
2	INT16	src-y
2	INT16	dst-x
2	INT16	dst-y
2	CARD16	width
2	CARD16	height

CopyPlane

1	63	opcode
1		unused
2	8	request length
4	DRAWABLE	src-drawable
4	DRAWABLE	dst-drawable
4	GCONTEXT	gc
2	INT16	src-x
2	INT16	src-y
2	INT16	dst-x
2	INT16	dst-y
2	CARD16	width
2	CARD16	height
4	CARD32	bit-plane

PolyPoint

1	64		opcode
1			coordinate-mode
	0	Origin	
	1	Previous	
2	3+n		request length
4	DRAWABLE		drawable
4	GCONTEXT		gc
4n	LISTofPOINT		points

PolyLine

1	65		opcode
1			coordinate-mode
	0	Origin	
	1	Previous	
2	3+n		request length
4	DRAWABLE		drawable
4	GCONTEXT		gc
4n	LISTofPOINT		points

PolySegment

1	66	opcode
1		unused
2	3+2n	request length
4	DRAWABLE	drawable
4	GCONTEXT	gc
8n	LISTofSEGMENT	segments

SEGMENT

2	INT16	x1
2	INT16	y1
2	INT16	x2
2	INT16	y2

PolyRectangle

1	67	opcode
1		unused
2	3+2n	request length
4	DRAWABLE	drawable
4	GCONTEXT	gc
8n	LISTofRECTANGLE	rectangles

PolyArc

1	68	opcode
1		unused
2	3+3n	request length
4	DRAWABLE	drawable
4	GCONTEXT	gc
12n	LISTofARC	arcs

FillPoly

1	69		opcode
1			unused
2	4+n		request length
4	DRAWABLE		drawable
4	GCONTEXT		gc
1			shape
	0	Complex	
	1	Nonconvex	
	2	Convex	
1			coordinate-mode
	0	Origin	
	1	Previous	
2			unused
4n	LISTofPOINT		points

PolyFillRectangle

1	70		opcode
1			unused
2	3+2n		request length
4	DRAWABLE		drawable
4	GCONTEXT		gc
8n	LISTofRECTANGLE		rectangles

PolyFillArc

1	71		opcode
1			unused
2	3+3n		request length
4	DRAWABLE		drawable
4	GCONTEXT		gc
12n	LISTofARC		arcs

PutImage

1	72		opcode
1			format
	0	Bitmap	
	1	XYPixmap	
	2	ZPixmap	
2	6+(n+p)/4		request length
4	DRAWABLE		drawable
4	GCONTEXT		gc
2	CARD16		width
2	CARD16		height
2	INT16		dst-x
2	INT16		dst-y
1	CARD8		left-pad
1	CARD8		depth
2			unused
n	LISTofBYTE		data
p			unused, p=pad(n)

GetImage

1	73		opcode
1			format
	1	XYPixmap	
	2	ZPixmap	
2	5		request length
4	DRAWABLE		drawable
2	INT16		x
2	INT16		y
2	CARD16		width
2	CARD16		height
4	CARD32		plane-mask

=>

1	1		Reply

1	CARD8	depth
2	CARD16	sequence number
4	(n+p)/4	reply length
4	VISUALID	visual
	0 None	
20		unused
n	LISTofBYTE	data
p		unused, p=pad(n)

PolyText8

1	74	opcode
1		unused
2	4+(n+p)/4	request length
4	DRAWABLE	drawable
4	GCONTEXT	gc
2	INT16	x
2	INT16	y
n	LISTofTEXTITEM8	items
p		unused, p=pad(n) (p is always 0 or 1)

TEXTITEM8

1	m	length of string (cannot be 255)
1	INT8	delta
m	STRING8	string
or		
1	255	font-shift indicator
1		font byte 3 (most-significant)
1		font byte 2
1		font byte 1
1		font byte 0 (least-significant)

PolyText16

1	75	opcode
1		unused
2	4+(n+p)/4	request length
4	DRAWABLE	drawable
4	GCONTEXT	gc
2	INT16	x
2	INT16	y
n	LISTofTEXTITEM16	items
p		unused, p=pad(n) (p must be 0 or 1)

TEXTITEM16

1	m	number of CHAR2Bs in string (cannot be 255)
1	INT8	delta
m	STRING16	string
or		
1	255	font-shift indicator
1		font byte 3 (most-significant)
1		font byte 2
1		font byte 1
1		font byte 0 (least-significant)

ImageText8

1	76	opcode
1	n	length of string
2	4+(n+p)/4	request length
4	DRAWABLE	drawable
4	GCONTEXT	gc
2	INT16	x
2	INT16	y
n	STRING8	string
p		unused, p=pad(n)

ImageText16

1	77	opcode

1	n		number of CHAR2Bs in string
2	4+(2n+p)/4		request length
4	DRAWABLE		drawable
4	GCONTEXT		gc
2	INT16		x
2	INT16		y
2n	STRING16		string
p			unused, p=pad(2n)

CreateColormap

1	78		opcode
1			alloc
	0	None	
	1	All	
2	4		request length
4	COLORMAP		mid
4	WINDOW		window
4	VISUALID		visual

FreeColormap

1	79	opcode
1		unused
2	2	request length
4	COLORMAP	cmap

CopyColormapAndFree

1	80	opcode
1		unused
2	3	request length
4	COLORMAP	mid
4	COLORMAP	src-cmap

InstallColormap

1	81	opcode
1		unused
2	2	request length
4	COLORMAP	cmap

UninstallColormap

1	82	opcode
1		unused
2	2	request length
4	COLORMAP	cmap

ListInstalledColormaps

1	83	opcode
1		unused
2	2	request length
4	WINDOW	window

=>

1	1	Reply
1		unused
2	CARD16	sequence number
4	n	reply length
2	n	number of COLORMAPs in cmaps
22		unused
4n	LISTofCOLORMAP	cmaps

AllocColor

1	84	opcode
1		unused
2	4	request length
4	COLORMAP	cmap
2	CARD16	red
2	CARD16	green

2	CARD16	blue
2		unused

=>

1	1	Reply
1		unused
2	CARD16	sequence number
4	0	reply length
2	CARD16	red
2	CARD16	green
2	CARD16	blue
2		unused
4	CARD32	pixel
12		unused

AllocNamedColor

1	85	opcode
1		unused
2	3+(n+p)/4	request length
4	COLORMAP	cmap
2	n	length of name
2		unused
n	STRING8	name
p		unused, p=pad(n)

=>

1	1	Reply
1		unused
2	CARD16	sequence number
4	0	reply length
4	CARD32	pixel
2	CARD16	exact-red
2	CARD16	exact-green
2	CARD16	exact-blue
2	CARD16	visual-red
2	CARD16	visual-green
2	CARD16	visual-blue
8		unused

AllocColorCells

1	86	opcode
1	BOOL	contiguous
2	3	request length
4	COLORMAP	cmap
2	CARD16	colors
2	CARD16	planes

=>

1	1	Reply
1		unused
2	CARD16	sequence number
4	n+m	reply length
2	n	number of CARD32s in pixels
2	m	number of CARD32s in masks
20		unused
4n	LISTofCARD32	pixels
4m	LISTofCARD32	masks

AllocColorPlanes

1	87	opcode
1	BOOL	contiguous
2	4	request length
4	COLORMAP	cmap
2	CARD16	colors
2	CARD16	reds
2	CARD16	greens
2	CARD16	blues

=>

1	1	Reply
1		unused
2	CARD16	sequence number
4	n	reply length
2	n	number of CARD32s in pixels
2		unused
4	CARD32	red-mask
4	CARD32	green-mask
4	CARD32	blue-mask
8		unused
4n	LISTofCARD32	pixels

FreeColors

1	88	opcode
1		unused
2	3+n	request length
4	COLORMAP	cmap
4	CARD32	plane-mask
4n	LISTofCARD32	pixels

StoreColors

1	89	opcode
1		unused
2	2+3n	request length
4	COLORMAP	cmap
12n	LISTofCOLORITEM	items

COLORITEM

4	CARD32	pixel
2	CARD16	red
2	CARD16	green
2	CARD16	blue
1		do-red, do-green, do-blue
	#x01	do-red (1 is True, 0 is False)
	#x02	do-green (1 is True, 0 is False)
	#x04	do-blue (1 is True, 0 is False)
	#xf8	unused
1		unused

StoreNamedColor

1	90	opcode
1		do-red, do-green, do-blue
	#x01	do-red (1 is True, 0 is False)
	#x02	do-green (1 is True, 0 is False)
	#x04	do-blue (1 is True, 0 is False)
	#xf8	unused
2	4+(n+p)/4	request length
4	COLORMAP	cmap
4	CARD32	pixel
2	n	length of name
2		unused
n	STRING8	name
p		unused, p=pad(n)

QueryColors

1	91	opcode
1		unused
2	2+n	request length
4	COLORMAP	cmap
4n	LISTofCARD32	pixels

=>

1	1	Reply
1		unused
2	CARD16	sequence number

4	2n	reply length	
2	n		number of RGBs in colors
22			unused
8n	LISTofRGB		colors

RGB

2	CARD16		red
2	CARD16		green
2	CARD16		blue
2			unused

LookupColor

1	92		opcode
1			unused
2	3+(n+p)/4		request length
4	COLORMAP		cmap
2	n		length of name
2			unused
n	STRING8		name
p			unused, p=pad(n)

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	0		reply length
2	CARD16		exact-red
2	CARD16		exact-green
2	CARD16		exact-blue
2	CARD16		visual-red
2	CARD16		visual-green
2	CARD16		visual-blue
12			unused

CreateCursor

1	93		opcode
1			unused
2	8		request length
4	CURSOR		cid
4	PIXMAP		source
4	PIXMAP		mask
	0	None	
2	CARD16		fore-red
2	CARD16		fore-green
2	CARD16		fore-blue
2	CARD16		back-red
2	CARD16		back-green
2	CARD16		back-blue
2	CARD16		x
2	CARD16		y

CreateGlyphCursor

1	94		opcode
1			unused
2	8		request length
4	CURSOR		cid
4	FONT		source-font
4	FONT		mask-font
	0	None	
2	CARD16		source-char
2	CARD16		mask-char
2	CARD16		fore-red
2	CARD16		fore-green
2	CARD16		fore-blue
2	CARD16		back-red
2	CARD16		back-green

2	CARD16	back-blue

FreeCursor

1	95	opcode
1		unused
2	2	request length
4	CURSOR	cursor

RecolorCursor

1	96	opcode
1		unused
2	5	request length
4	CURSOR	cursor
2	CARD16	fore-red
2	CARD16	fore-green
2	CARD16	fore-blue
2	CARD16	back-red
2	CARD16	back-green
2	CARD16	back-blue

QueryBestSize

1	97		opcode
1			class
	0	Cursor	
	1	Tile	
	2	Stipple	
2	3		request length
4	DRAWABLE		drawable
2	CARD16		width
2	CARD16		height

=>

1	1	Reply
1		unused
2	CARD16	sequence number
4	0	reply length
2	CARD16	width
2	CARD16	height
20		unused

QueryExtension

1	98	opcode
1		unused
2	2+(n+p)/4	request length
2	n	length of name
2		unused
n	STRING8	name
p		unused, p=pad(n)

=>

1	1	Reply
1		unused
2	CARD16	sequence number
4	0	reply length
1	BOOL	present
1	CARD8	major-opcode
1	CARD8	first-event
1	CARD8	first-error
20		unused

ListExtensions

1	99	opcode
1		unused
2	1	request length

=>

1	1	Reply

1	CARD8		number of STRs in names
2	CARD16		sequence number
4	(n+p)/4		reply length
24			unused
n	LISTofSTR		names
p			unused, p=pad(n)

ChangeKeyboardMapping

1	100		opcode
1	n		keycode-count
2	2+nm		request length
1	KEYCODE		first-keycode
1	m		keysyms-per-keycode
2			unused
4nm	LISTofKEYSYM		keysyms

GetKeyboardMapping

1	101		opcode
1			unused
2	2		request length
1	KEYCODE		first-keycode
1	m		count
2			unused

=>

1	1		Reply
1	n		keysyms-per-keycode
2	CARD16		sequence number
4	nm		reply length (m = count field from the request)
24			unused
4nm	LISTofKEYSYM		keysyms

ChangeKeyboardControl

1	102		opcode
1			unused
2	2+n		request length
4	BITMASK		value-mask (has n bits set to 1)
	#x0001	key-click-percent	
	#x0002	bell-percent	
	#x0004	bell-pitch	
	#x0008	bell-duration	
	#x0010	led	
	#x0020	led-mode	
	#x0040	key	
	#x0080	auto-repeat-mode	
4n	LISTofVALUE		value-list

VALUEs

1	INT8		key-click-percent
1	INT8		bell-percent
2	INT16		bell-pitch
2	INT16		bell-duration
1	CARD8		led
1			led-mode
	0	Off	
	1	On	
1	KEYCODE		key
1			auto-repeat-mode
	0	Off	
	1	On	
	2	Default	

GetKeyboardControl

1	103		opcode
1			unused
2	1		request length

=>

1	1		Reply
1			global-auto-repeat
	0	Off	
	1	On	
2	CARD16		sequence number
4	5		reply length
4	CARD32		led-mask
1	CARD8		key-click-percent
1	CARD8		bell-percent
2	CARD16		bell-pitch
2	CARD16		bell-duration
2			unused
32	LISTofCARD8		auto-repeats

Bell

1	104	opcode
1	INT8	percent
2	1	request length

ChangePointerControl

1	105	opcode
1		unused
2	3	request length
2	INT16	acceleration-numerator
2	INT16	acceleration-denominator
2	INT16	threshold
1	BOOL	do-acceleration
1	BOOL	do-threshold

GetPointerControl

1	106	opcode
1		unused
2	1	request length

=>

1	1	Reply
1		unused
2	CARD16	sequence number
4	0	reply length
2	CARD16	acceleration-numerator
2	CARD16	acceleration-denominator
2	CARD16	threshold
18		unused

SetScreenSaver

1	107		opcode
1			unused
2	3		request length
2	INT16		timeout
2	INT16		interval
1			prefer-blanking
	0	No	
	1	Yes	
	2	Default	
1			allow-exposures
	0	No	
	1	Yes	
	2	Default	
2			unused

GetScreenSaver

1	108	opcode
1		unused
2	1	request length

=>

1	1		Reply
1			unused
2	CARD16		sequence number
4	0		reply length
2	CARD16		timeout
2	CARD16		interval
1			prefer-blanking
	0	No	
	1	Yes	
1			allow-exposures
	0	No	
	1	Yes	
18			unused

ChangeHosts

1	109		opcode
1			mode
	0	Insert	
	1	Delete	
2	2+(n+p)/4		request length
1			family
	0	Internet	
	1	DECnet	
	2	Chaos	
1			unused
2	n		length of address
n	LISTofCARD8		address
p			unused, p=pad(n)

ListHosts

1	110		opcode
1			unused
2	1		request length

=>

1	1		Reply
1			mode
	0	Disabled	
	1	Enabled	
2	CARD16		sequence number
4	n/4		reply length
2	CARD16		number of HOSTs in hosts
22			unused
n	LISTofHOST		hosts (n always a multiple of 4)

SetAccessControl

1	111		opcode
1			mode
	0	Disable	
	1	Enable	
2	1		request length

SetCloseDownMode

1	112		opcode
1			mode
	0	Destroy	
	1	RetainPermanent	
	2	RetainTemporary	
2	1		request length

KillClient

1	113		opcode
1			unused
2	2		request length
4	CARD32		resource

0	AllTemporary		

RotateProperties

1	114		opcode
1			unused
2	3+n		request length
4	WINDOW		window
2	n		number of properties
2	INT16		delta
4n	LISTofATOM		properties

ForceScreenSaver

1	115		opcode
1			mode
	0	Reset	
	1	Activate	
2	1		request length

SetPointerMapping

1	116		opcode
1	n		length of map
2	1+(n+p)/4		request length
n	LISTofCARD8		map
p			unused, p=pad(n)

=>

1	1		Reply
1			status
	0	Success	
	1	Busy	
2	CARD16		sequence number
4	0		reply length
24			unused

GetPointerMapping

1	117		opcode
1			unused
2	1		request length

=>

1	1		Reply
1	n		length of map
2	CARD16		sequence number
4	(n+p)/4		reply length
24			unused
n	LISTofCARD8		map
p			unused, p=pad(n)

SetModifierMapping

1	118		opcode
1	n		keycodes-per-modifier
2	1+2n		request length
8n	LISTofKEYCODE		keycodes

=>

1	1		Reply
1			status
	0	Success	
	1	Busy	
	2	Failed	
2	CARD16		sequence number
4	0		reply length
24			unused

GetModifierMapping

1	119		opcode
1			unused

2 1 request length

=>

1	1	Reply
1	n	keycodes-per-modifier
2	CARD16	sequence number
4	2n	reply length
24		unused
8n	LISTofKEYCODE	keycodes

NoOperation

1	127	opcode
1		unused
2	1	request length

Events

KeyPress

1	2	code
1	KEYCODE	detail
2	CARD16	sequence number
4	TIMESTAMP	time
4	WINDOW	root
4	WINDOW	event
4	WINDOW	child
	0 None	
2	INT16	root-x
2	INT16	root-y
2	INT16	event-x
2	INT16	event-y
2	SETofKEYBUTMASK	state
1	BOOL	same-screen
1		unused

KeyRelease

1	3	code
1	KEYCODE	detail
2	CARD16	sequence number
4	TIMESTAMP	time
4	WINDOW	root
4	WINDOW	event
4	WINDOW	child
	0 None	
2	INT16	root-x
2	INT16	root-y
2	INT16	event-x
2	INT16	event-y
2	SETofKEYBUTMASK	state
1	BOOL	same-screen
1		unused

ButtonPress

1	4	code
1	BUTTON	detail
2	CARD16	sequence number
4	TIMESTAMP	time
4	WINDOW	root
4	WINDOW	event
4	WINDOW	child
	0 None	
2	INT16	root-x
2	INT16	root-y
2	INT16	event-x
2	INT16	event-y
2	SETofKEYBUTMASK	state
1	BOOL	same-screen

1		unused	

ButtonRelease

1	5		code
1	BUTTON		detail
2	CARD16		sequence number
4	TIMESTAMP		time
4	WINDOW		root
4	WINDOW		event
4	WINDOW		child
	0	None	
2	INT16		root-x
2	INT16		root-y
2	INT16		event-x
2	INT16		event-y
2	SETofKEYBUTMASK		state
1	BOOL		same-screen
1			unused

MotionNotify

1	6		code
1			detail
	0	Normal	
	1	Hint	
2	CARD16		sequence number
4	TIMESTAMP		time
4	WINDOW		root
4	WINDOW		event
4	WINDOW		child
	0	None	
2	INT16		root-x
2	INT16		root-y
2	INT16		event-x
2	INT16		event-y
2	SETofKEYBUTMASK		state
1	BOOL		same-screen
1			unused

EnterNotify

1	7		code
1			detail
	0	Ancestor	
	1	Virtual	
	2	Inferior	
	3	Nonlinear	
	4	NonlinearVirtual	
2	CARD16		sequence number
4	TIMESTAMP		time
4	WINDOW		root
4	WINDOW		event
4	WINDOW		child
	0	None	
2	INT16		root-x
2	INT16		root-y
2	INT16		event-x
2	INT16		event-y
2	SETofKEYBUTMASK		state
1			mode
	0	Normal	
	1	Grab	
	2	Ungrab	
1			same-screen, focus
	#x01	focus (1 is True, 0 is False)	
	#x02	same-screen (1 is True, 0 is False)	
	#xfc	unused	

LeaveNotify

1	8		code
1			detail
	0	Ancestor	
	1	Virtual	
	2	Inferior	
	3	Nonlinear	
	4	NonlinearVirtual	
2	CARD16		sequence number
4	TIMESTAMP		time
4	WINDOW		root
4	WINDOW		event
4	WINDOW		child
	0	None	
2	INT16		root-x
2	INT16		root-y
2	INT16		event-x
2	INT16		event-y
2	SETofKEYBUTMASK		state
1			mode
	0	Normal	
	1	Grab	
	2	Ungrab	
1			same-screen, focus
	#x01	focus (1 is True, 0 is False)	
	#x02	same-screen (1 is True, 0 is False)	
	#xfc	unused	

FocusIn

1	9		code
1			detail
	0	Ancestor	
	1	Virtual	
	2	Inferior	
	3	Nonlinear	
	4	NonlinearVirtual	
	5	Pointer	
	6	PointerRoot	
	7	None	
2	CARD16		sequence number
4	WINDOW		event
1			mode
	0	Normal	
	1	Grab	
	2	Ungrab	
	3	WhileGrabbed	
23			unused

FocusOut

1	10		code
1			detail
	0	Ancestor	
	1	Virtual	
	2	Inferior	
	3	Nonlinear	
	4	NonlinearVirtual	
	5	Pointer	
	6	PointerRoot	
	7	None	
2	CARD16		sequence number
4	WINDOW		event
1			mode
	0	Normal	
	1	Grab	
	2	Ungrab	

3	WhileGrabbed	
23		unused

KeymapNotify

1	11	code
31	LISTofCARD8	keys (byte for keycodes 0–7 is omitted)

Expose

1	12	code
1		unused
2	CARD16	sequence number
4	WINDOW	window
2	CARD16	x
2	CARD16	y
2	CARD16	width
2	CARD16	height
2	CARD16	count
14		unused

GraphicsExposure

1	13	code
1		unused
2	CARD16	sequence number
4	DRAWABLE	drawable
2	CARD16	x
2	CARD16	y
2	CARD16	width
2	CARD16	height
2	CARD16	minor-opcode
2	CARD16	count
1	CARD8	major-opcode
11		unused

NoExposure

1	14	code
1		unused
2	CARD16	sequence number
4	DRAWABLE	drawable
2	CARD16	minor-opcode
1	CARD8	major-opcode
21		unused

VisibilityNotify

1	15		code
1			unused
2	CARD16		sequence number
4	WINDOW		window
1			state
	0	Unobscured	
	1	PartiallyObscured	
	2	FullyObscured	
23			unused

CreateNotify

1	16	code
1		unused
2	CARD16	sequence number
4	WINDOW	parent
4	WINDOW	window
2	INT16	x
2	INT16	y
2	CARD16	width
2	CARD16	height
2	CARD16	border-width
1	BOOL	override-redirect
9		unused

DestroyNotify

1	17		code
1			unused
2	CARD16		sequence number
4	WINDOW		event
4	WINDOW		window
20			unused

UnmapNotify

1	18		code
1			unused
2	CARD16		sequence number
4	WINDOW		event
4	WINDOW		window
1	BOOL		from-configure
19			unused

MapNotify

1	19		code
1			unused
2	CARD16		sequence number
4	WINDOW		event
4	WINDOW		window
1	BOOL		override-redirect
19			unused

MapRequest

1	20		code
1			unused
2	CARD16		sequence number
4	WINDOW		parent
4	WINDOW		window
20			unused

ReparentNotify

1	21		code
1			unused
2	CARD16		sequence number
4	WINDOW		event
4	WINDOW		window
4	WINDOW		parent
2	INT16		x
2	INT16		y
1	BOOL		override-redirect
11			unused

ConfigureNotify

1	22		code
1			unused
2	CARD16		sequence number
4	WINDOW		event
4	WINDOW		window
4	WINDOW		above-sibling
	0	None	
2	INT16		x
2	INT16		y
2	CARD16		width
2	CARD16		height
2	CARD16		border-width
1	BOOL		override-redirect
5			unused

ConfigureRequest

1	23		code
1			stack-mode
	0	Above	

1	Below		
	2	TopIf	
	3	BottomIf	
	4	Opposite	
2	CARD16		sequence number
4	WINDOW		parent
4	WINDOW		window
4	WINDOW		sibling
	0	None	
2	INT16		x
2	INT16		y
2	CARD16		width
2	CARD16		height
2	CARD16		border-width
2	BITMASK		value-mask
	#x0001	x	
	#x0002	y	
	#x0004	width	
	#x0008	height	
	#x0010	border-width	
	#x0020	sibling	
	#x0040	stack-mode	
4			unused

GravityNotify

1	24	code
1		unused
2	CARD16	sequence number
4	WINDOW	event
4	WINDOW	window
2	INT16	x
2	INT16	y
16		unused

ResizeRequest

1	25	code
1		unused
2	CARD16	sequence number
4	WINDOW	window
2	CARD16	width
2	CARD16	height
20		unused

CirculateNotify

1	26		code
1			unused
2	CARD16		sequence number
4	WINDOW		event
4	WINDOW		window
4	WINDOW		unused
1			place
	0	Top	
	1	Bottom	
15			unused

CirculateRequest

1	27		code
1			unused
2	CARD16		sequence number
4	WINDOW		parent
4	WINDOW		window
4			unused
1			place
	0	Top	
	1	Bottom	

15		unused	

PropertyNotify

1	28		code
1			unused
2	CARD16		sequence number
4	WINDOW		window
4	ATOM		atom
4	TIMESTAMP		time
1			state
	0	NewValue	
	1	Deleted	
15			unused

SelectionClear

1	29		code
1			unused
2	CARD16		sequence number
4	TIMESTAMP		time
4	WINDOW		owner
4	ATOM		selection
16			unused

SelectionRequest

1	30		code
1			unused
2	CARD16		sequence number
4	TIMESTAMP		time
	0	CurrentTime	
4	WINDOW		owner
4	WINDOW		requestor
4	ATOM		selection
4	ATOM		target
4	ATOM		property
	0	None	
4			unused

SelectionNotify

1	31		code
1			unused
2	CARD16		sequence number
4	TIMESTAMP		time
	0	CurrentTime	
4	WINDOW		requestor
4	ATOM		selection
4	ATOM		target
4	ATOM		property
	0	None	
8			unused

ColormapNotify

1	32		code
1			unused
2	CARD16		sequence number
4	WINDOW		window
4	COLORMAP		colormap
	0	None	
1	BOOL		new
1			state
	0	Uninstalled	
	1	Installed	
18			unused

ClientMessage

1	33		code
1	CARD8		format

2	CARD16		sequence number
4	WINDOW		window
4	ATOM		type
20			data

MappingNotify

1	34		code
1			unused
2	CARD16		sequence number
1			request
	0	Modifier	
	1	Keyboard	
	2	Pointer	
1	KEYCODE		first-keycode
1	CARD8		count
25			unused

Glossary

Access control list

X maintains a list of hosts from which client programs can be run. By default, only programs on the local host and hosts specified in an initial list read by the server can use the display. Clients on the local host can change this access control list. Some server implementations can also implement other authorization mechanisms in addition to or in place of this mechanism. The action of this mechanism can be conditional based on the authorization protocol name and data received by the server at connection setup.

Active grab

A grab is active when the pointer or keyboard is actually owned by the single grabbing client.

Ancestors

If W is an inferior of A, then A is an ancestor of W.

Atom

An atom is a unique ID corresponding to a string name. Atoms are used to identify properties, types, and selections.

Background

An **InputOutput** window can have a background, which is defined as a pixmap. When regions of the window have their contents lost or invalidated, the server will automatically tile those regions with the background.

Backing store

When a server maintains the contents of a window, the pixels saved off screen are known as a backing store.

Bit gravity

When a window is resized, the contents of the window are not necessarily discarded. It is possible to request that the server relocate the previous contents to some region of the window (though no guarantees are made). This attraction of window contents for some location of a window is known as bit gravity.

Bit plane

When a pixmap or window is thought of as a stack of bitmaps, each bitmap is called a bit plane or plane.

Bitmap

A bitmap is a pixmap of depth one.

Border

An **InputOutput** window can have a border of equal thickness on all four sides of the window. A pixmap defines the contents of the border, and the server automatically maintains the contents of the border. Exposure events are never generated for border regions.

Button grabbing

Buttons on the pointer may be passively grabbed by a client. When the button is pressed, the pointer is then actively grabbed by the client.

Byte order

For image (pixmap/bitmap) data, the server defines the byte order, and clients with different native byte ordering must swap bytes as necessary. For all other parts of the protocol, the client defines the byte order, and the server swaps bytes as necessary.

Children

The children of a window are its first-level subwindows.

Client

An application program connects to the window system server by some interprocess communication (IPC) path, such as a TCP connection or a shared memory buffer. This program is referred to as a client of the window system server. More precisely, the client is the IPC path itself; a program with multiple paths open to the server is viewed as multiple clients by the protocol. Resource lifetimes are controlled by connection lifetimes, not by program lifetimes.

Clipping region

In a graphics context, a bitmap or list of rectangles can be specified to restrict output to a particular region of the window. The image defined by the bitmap or rectangles is called a clipping region.

Colormap

A colormap consists of a set of entries defining color values. The colormap associated with a window is used to display the contents of the window; each pixel value indexes the colormap to produce RGB values that drive the guns of a monitor. Depending on hardware limitations, one or more colormaps may be installed at one time, so that windows associated with those maps display with correct colors.

Connection

The IPC path between the server and client program is known as a connection. A client program typically (but not necessarily) has one connection to the server over which requests and events are sent.

Containment

A window ‘‘contains’’ the pointer if the window is viewable and the hotspot of the cursor is within a visible region of the window or a visible region of one of its inferiors. The border of the window is included as part of the window for containment. The pointer is ‘‘in’’ a window if the window contains the pointer but no inferior contains the pointer.

Coordinate system

The coordinate system has X horizontal and Y vertical, with the origin [0, 0] at the upper left. Coordinates are integral and coincide with pixel centers. Each window and pixmap has its own coordinate system. For a window, the origin is inside the border at the inside upper left.

Cursor

A cursor is the visible shape of the pointer on a screen. It consists of a hot spot, a source bitmap, a shape bitmap, and a pair of colors. The cursor defined for a window controls the visible appearance when the pointer is in that window.

Depth

The depth of a window or pixmap is the number of bits per pixel that it has. The depth of a graphics context is the depth of the drawables it can be used in conjunction with for graphics output.

Device

Keyboards, mice, tablets, track-balls, button boxes, and so on are all collectively known as input devices. The core protocol only deals with two devices, ‘‘the keyboard’’ and ‘‘the pointer.’’

DirectColor

DirectColor is a class of colormap in which a pixel value is decomposed into three separate subfields for indexing. The first subfield indexes an array to produce red intensity values. The second subfield indexes a second array to produce blue intensity values. The third subfield indexes a third array to produce green intensity values. The RGB values can be changed dynamically.

Display

A server, together with its screens and input devices, is called a display.

Drawable

Both windows and pixmaps can be used as sources and destinations in graphics operations. These windows and pixmaps are collectively known as drawables. However, an **InputOnly** window cannot be used as a source or destination in a graphics operation.

Event

Clients are informed of information asynchronously by means of events. These events can be generated either asynchronously from devices or as side effects of client requests. Events are grouped into types. The server never sends events to a client unless the client has specificially asked to be informed of that type of event. However, other clients can force events to be sent to other clients. Events are typically reported relative to a window.

Event mask

Events are requested relative to a window. The set of event types that a client requests relative to a window is described by using an event mask.

Event synchronization

There are certain race conditions possible when demultiplexing device events to clients (in particular deciding where pointer and keyboard events should be sent when in the middle of window management operations). The event synchronization mechanism allows synchronous processing of device events.

Event propagation

Device-related events propagate from the source window to ancestor windows until some client has expressed interest in handling that type of event or until the event is discarded explicitly.

Event source

The window the pointer is in is the source of a device-related event.

Exposure event

Servers do not guarantee to preserve the contents of windows when windows are obscured or reconfigured. Exposure events are sent to clients to inform them when contents of regions of windows have been lost.

Extension

Named extensions to the core protocol can be defined to extend the system. Extension to output requests, resources, and event types are all possible and are expected.

Focus window

The focus window is another term for the input focus.

Font

A font is a matrix of glyphs (typically characters). The protocol does no translation or interpretation of character sets. The client simply indicates values used to index the glyph array. A font contains additional metric information to determine interglyph and interline spacing.

GC, **GContext**

GC and gcontext are abbreviations for graphics context.

Glyph

A glyph is an image, typically of a character, in a font.

Grab

Keyboard keys, the keyboard, pointer buttons, the pointer, and the server can be grabbed for exclusive use by a client. In general, these facilities are not intended to be used by normal applications but are intended for various input and window managers to implement various styles of user interfaces.

Graphics context

Various information for graphics output is stored in a graphics context such as foreground pixel, background pixel, line width, clipping region, and so on. A graphics context can only be used with drawables that have the same root and the same depth as the graphics context.

Gravity

See **bit gravity** and **window gravity**.

GrayScale

GrayScale can be viewed as a degenerate case of **PseudoColor**, in which the red, green, and blue values in any given colormap entry are equal, thus producing shades of gray. The gray values can be changed dynamically.

Hotspot

A cursor has an associated hotspot that defines the point in the cursor corresponding to the coordinates reported for the pointer.

Identifier

An identifier is a unique value associated with a resource that clients use to name that resource. The identifier can be used over any connection.

Inferiors

The inferiors of a window are all of the subwindows nested below it: the children, the children's children, and so on.

Input focus

The input focus is normally a window defining the scope for processing of keyboard input. If a generated keyboard event would normally be reported to this window or one of its inferiors, the event is reported normally. Otherwise, the event is reported with respect to the focus window. The input focus also can be set such that all keyboard events are discarded and such that the focus window is dynamically taken to be the root window of whatever screen the pointer is on at each keyboard event.

Input manager

Control over keyboard input is typically provided by an input manager client.

InputOnly window

An **InputOnly** window is a window that cannot be used for graphics requests. **InputOnly** windows are invisible and can be used to control such things as cursors, input event generation, and grabbing. **InputOnly** windows cannot have **InputOutput** windows as inferiors.

InputOutput window

An **InputOutput** window is the normal kind of opaque window, used for both input and output. **InputOutput** windows can have both **InputOutput** and **InputOnly** windows as inferiors.

Key grabbing

Keys on the keyboard can be passively grabbed by a client. When the key is pressed, the keyboard is then actively grabbed by the client.

Keyboard grabbing

A client can actively grab control of the keyboard, and key events will be sent to that client rather than the client the events would normally have been sent to.

Keysym

An encoding of a symbol on a keycap on a keyboard.

Mapped

A window is said to be mapped if a map call has been performed on it. Unmapped windows and their inferiors are never viewable or visible.

Modifier keys

Shift, Control, Meta, Super, Hyper, Alt, Compose, Apple, CapsLock, ShiftLock, and similar keys are called modifier keys.

Monochrome

Monochrome is a special case of **StaticGray** in which there are only two colormap entries.

Obscure

A window is obscured if some other window obscures it. Window A obscures window B if both are viewable **InputOutput** windows, A is higher in the global stacking order, and the rectangle defined by the outside edges of A intersects the rectangle defined by the outside edges of B. Note the distinction between obscure and occludes. Also note that window borders are included in the calculation and that a window can be obscured and yet still have visible regions.

Occlude

A window is occluded if some other window occludes it. Window A occludes window B if both are mapped, A is higher in the global stacking order, and the rectangle defined by the outside edges of A intersects the rectangle defined by the outside edges of B. Note the distinction between occludes and obscures. Also note that window borders are included in the calculation.

Padding

Some padding bytes are inserted in the data stream to maintain alignment of the protocol requests on natural boundaries. This increases ease of portability to some machine architectures.

Parent window

If C is a child of P, then P is the parent of C.

Passive grab

Grabbing a key or button is a passive grab. The grab activates when the key or button is actually pressed.

Pixel value

A pixel is an N-bit value, where N is the number of bit planes used in a particular window or pixmap (that is, N is the depth of the window or pixmap). For a window, a pixel value indexes a colormap to derive an actual color to be displayed.

Pixmap

A pixmap is a three-dimensional array of bits. A pixmap is normally thought of as a two-dimensional array of pixels, where each pixel can be a value from 0 to (2ˆN)-1 and where N is the depth (z axis) of the pixmap. A pixmap can also be thought of as a stack of N bitmaps.

Plane

When a pixmap or window is thought of as a stack of bitmaps, each bitmap is called a plane or bit plane.

Plane mask

Graphics operations can be restricted to only affect a subset of bit planes of a destination. A plane mask is a bit mask describing which planes are to be modified. The plane mask is stored in a graphics context.

Pointer

The pointer is the pointing device attached to the cursor and tracked on the screens.

Pointer grabbing

A client can actively grab control of the pointer. Then button and motion events will be sent to that client rather than the client the events would normally have been sent to.

Pointing device

A pointing device is typically a mouse, tablet, or some other device with effective dimensional motion. There is only one visible cursor defined by the core protocol, and it tracks whatever pointing device is attached as the pointer.

Property

Windows may have associated properties, which consist of a name, a type, a data format, and some data. The protocol places no interpretation on properties. They are intended as a general-purpose naming mechanism for clients. For example, clients might use properties to share information such as resize hints, program names, and icon formats with a window manager.

Property list

The property list of a window is the list of properties that have been defined for the window.

PseudoColor

PseudoColor is a class of colormap in which a pixel value indexes the colormap to produce independent red, green, and blue values; that is, the colormap is viewed as an array of triples (RGB values). The RGB values can be changed dynamically.

Redirecting control

Window managers (or client programs) may want to enforce window layout policy in various ways. When a client attempts to change the size or position of a window, the operation may be redirected to a specified client rather than the operation actually being performed.

Reply

Information requested by a client program is sent back to the client with a reply. Both events and replies are multiplexed on the same connection. Most requests do not generate replies, although some requests generate multiple replies.

Request

A command to the server is called a request. It is a single block of data sent over a connection.

Resource

Windows, pixmaps, cursors, fonts, graphics contexts, and colormaps are known as resources. They all have unique identifiers associated with them for naming purposes. The lifetime of a resource usually is bounded by the lifetime of the connection over which the resource was created.

RGB values

Red, green, and blue (RGB) intensity values are used to define color. These values are always represented as 16-bit unsigned numbers, with 0 being the minimum intensity and 65535 being the maximum intensity. The server scales the values to match the display hardware.

Root

The root of a pixmap or graphics context is the same as the root of whatever drawable was used when the pixmap or graphics context was created. The root of a window is the root window under which the window was created.

Root window

Each screen has a root window covering it. It cannot be reconfigured or unmapped, but it otherwise acts as a full-fledged window. A root window has no parent.

Save set

The save set of a client is a list of other clients’ windows that, if they are inferiors of one of the client’s windows at connection close, should not be destroyed and that should be remapped if currently unmapped. Save sets are typically used by window managers to avoid lost windows if the manager terminates abnormally.

Scanline

A scanline is a list of pixel or bit values viewed as a horizontal row (all values having the same y coordinate) of an image, with the values ordered by increasing x coordinate.

Scanline order

An image represented in scanline order contains scanlines ordered by increasing y coordinate.

Screen

A server can provide several independent screens, which typically have physically independent monitors. This would be the expected configuration when there is only a single keyboard and pointer shared among the screens.

Selection

A selection can be thought of as an indirect property with dynamic type; that is, rather than having the property stored in the server, it is maintained by some client (the ‘‘owner’’). A selection is global in nature and is thought of as belonging to the user (although maintained by clients), rather than as being private to a particular window subhierarchy or a particular set of clients. When a client asks for the contents of a selection, it specifies a selection ‘‘target type’’. This target type can be used to control the transmitted representation of the contents. For example, if the selection is ‘‘the last thing the user clicked on’’ and that is currently an image, then the target type might specify whether the contents of the image should be sent in XY format or Z format. The target type can also be used to control the class of contents transmitted; for example, asking for the ‘‘looks’’ (fonts, line spacing, indentation, and so on) of a paragraph selection rather than the text of the paragraph. The target type can also be used for other purposes. The protocol does not constrain the semantics.

Server

The server provides the basic windowing mechanism. It handles IPC connections from clients, multiplexes graphics requests onto the screens, and demultiplexes input back to the appropriate clients.

Server grabbing

The server can be grabbed by a single client for exclusive use. This prevents processing of any requests from other client connections until the grab is completed. This is typically only a transient state for such things as rubber-banding, pop-up menus, or to execute requests indivisibly.

Sibling

Children of the same parent window are known as sibling windows.

Stacking order

Sibling windows may stack on top of each other. Windows above other windows both obscure and occlude those lower windows. This is similar to paper on a desk. The relationship between sibling windows is known as the stacking order.

StaticColor

StaticColor can be viewed as a degenerate case of **PseudoColor** in which the RGB values are predefined and read-only.

StaticGray

StaticGray can be viewed as a degenerate case of **GrayScale** in which the gray values are predefined and read-only. The values are typically linear or near-linear increasing ramps.

Stipple

A stipple pattern is a bitmap that is used to tile a region that will serve as an additional clip mask for a fill operation with the foreground color.

String Equivalence

Two ISO Latin-1 STRING8 values are considered equal if they are the same length and if corresponding bytes are either equal or are equivalent as follows: decimal values 65 to 90 inclusive (characters ‘‘A’’ to ‘‘Z’’) are pairwise equivalent to decimal values 97 to 122 inclusive (characters ‘‘a’’ to ‘‘z’’), decimal values 192 to 214 inclusive (characters ‘‘A grave’’ to ‘‘O diaeresis’’) are pairwise equivalent to decimal values 224 to 246 inclusive (characters ‘‘a grave’’ to ‘‘o diaeresis’’), and decimal values 216 to 222 inclusive (characters ‘‘O oblique’’ to ‘‘THORN’’) are pairwise equivalent to decimal values 246 to 254 inclusive (characters ‘‘o oblique’’ to ‘‘thorn’’).

Tile

A pixmap can be replicated in two dimensions to tile a region. The pixmap itself is also known as a tile.

Timestamp

A timestamp is a time value, expressed in milliseconds. It typically is the time since the last server reset. Timestamp values wrap around (after about 49.7 days). The server, given its current time is represented by timestamp T, always interprets timestamps from clients by treating half of the timestamp space as being earlier in time than T and half of the timestamp space as being later in time than T. One timestamp value (named **CurrentTime**) is never generated by the server. This value is reserved for use in requests to represent the current server time.

TrueColor

TrueColor can be viewed as a degenerate case of **DirectColor** in which the subfields in the pixel value directly encode the corresponding RGB values; that is, the colormap has predefined read-only RGB values. The values are typically linear or near-linear increasing ramps.

Type

A type is an arbitrary atom used to identify the interpretation of property data. Types are completely uninterpreted by the server and are solely for the benefit of clients.

Viewable

A window is viewable if it and all of its ancestors are mapped. This does not imply that any portion of the window is actually visible. Graphics requests can be performed on a window when it is not viewable, but output will not be retained unless the server is maintaining backing store.

Visible

A region of a window is visible if someone looking at the screen can actually see it; that is, the window is viewable and the region is not occluded by any other window.

Window gravity

When windows are resized, subwindows may be repositioned automatically relative to some position in the window. This attraction of a subwindow to some part of its parent is known as window gravity.

Window manager

Manipulation of windows on the screen and much of the user interface (policy) is typically provided by a window manager client.

XYFormat

The data for a pixmap is said to be in XY format if it is organized as a set of bitmaps representing individual bit planes, with the planes appearing from most-significant to least-significant in bit order.

ZFormat

The data for a pixmap is said to be in Z format if it is organized as a set of pixel values in scanline order.