Attribute Characterization

SkyTel WCTP 1.1/WCTP 1.2 Beta Release 1

Document #S266 Revision 1.02

September 11, 2002

Table of Contents

1 Introduction

This document characterizes all attributes used by WCTP to include the format of the data, minimum and maximum data lengths, range of values, and any enumerated values. The **Section 2** defines the data types that will be used in the tables that describe the attributes for each of the WCTP elements. **Section 3** summarizes the defined types. Finally, **Section 4** includes tables for each element's attributes.

2 The Types

This document uses the "BNF-like format" as described in Appendix A (originally from RFC 2396, Section 1.6), to describe the valid formats for the fields. See Appendix A for full details on the grammar and usage rules.
Literal characters (characters that must appear exactly as shown) used in defining the format for the types are shown in **bold** type. Type names are shown in *italics*.

2.1 Basic Types

There are several basic types needed to describe the more complex types in WCTP. These basic types are defined here and are used in later sections.

2.1.1 Digit

Format: A *Digit*, representing a single numeric character, must conform to the following format:
Digit = "**0**" | "**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**" | "**7**" | "**8**" | "**9**"
Length: *Digits* always have a length of one character.
Range: The minimum equivalent integral value for a *Digit* is 0 and the maximum is 9.
Purpose: To represent a single numeric character for use in other more complex types.

2.1.2 LowAlpha

Format: A *LowAlpha*, representing a single lower-case alphabetic character, must conform to the following format:
LowAlpha = "**a**" | "**b**" | "**c**" | "**d**" | "**e**" | "**f**" | "**g**" | "**h**" | "**i**" | "**j**" | "**k**" | "**l**" | "**m**" | "**n**" | "**o**" | "**p**" | "**q**" | "**r**" | "**s**" | "**t**" | "**u**" | "**v**" | "**w**" | "**x**" | "**y**" | "**z**"
Length: *LowAlphas* always have a length of one character.
Purpose: To represent a single lower-case alphabetic character for use in other more complex types.

2.1.3 UpAlpha

Format: An *UpAlpha*, representing a single upper-case alphabetic character, must conform to the following format:
UpAlpha = "**A**" | "**B**" | "**C**" | "**D**" | "**E**" | "**F**" | "**G**" | "**H**" | "**I**" | "**J**" | "**K**" | "**L**" | "**M**" | "**N**" | "**O**" | "**P**" | "**Q**" | "**R**" | "**S**" | "**T**" | "**U**" | "**V**" | "**W**" | "**X**" | "**Y**" | "**Z**"
Length: *UpAlphas* always have a length of one character.
Purpose: To represent a single upper-case alphabetic character for use in other more complex types.

2.1.4 Alpha

Format: An *Alpha*, representing a single lower- or upper-case alphabetic character, must conform to the following format:
Alpha = LowAlpha |UpAlpha
Length: *Alphas* always have a length of one character.
Purpose: To represent a single upper-case alphabetic character for use in other more complex types.

2.1.5 AlphaNum

Format: An *AlphaNum*, representing a single alphanumeric character, must conform to the following format:
AlphaNum = Alpha |Digit
Length: *AlphaNums* always have a length of one character.
Purpose: To represent a single alphanumeric character in other more complex types.

2.1.6 NegativeNumber

Format: A *NegativeNumber*, a series of characters that together represent a negative (less than zero) integer value, must conform to the following format:
NegativeNumber = "**-**" ("**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**" | "**7**" | "**8**" | "**9**") **Digit*
Length: *NegativeNumbers* have a minimum length of two characters; the maximum length is unspecified, but should be stated for each attribute that uses this type.
Purpose: To define a negative integer type.

2.1.7 PositiveNumber

Format: A *PositiveNumber*, a series of characters that together represent a positive (greater than zero) integer value, must conform to the following format:
PositiveNumber = ("**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**" | "**7**" | "**8**" | "**9**") **Digit*
Length: *PositiveNumbers* have a minimum length of one character; the maximum length is unspecified, but should be stated for each attribute that uses this type.
Purpose: To define a positive integer type.

2.1.8 NonPositiveNumber

Format: A *NonPositiveNumber*, a series of characters that together represent a non-positive (less than or equal to zero) integer value, must conform to the following format:
NonPositiveNumber = NegativeNumber | "**0**"
Length: *NonPositiveNumbers* have a minimum length of one character; the maximum length is unspecified, but should be stated for each attribute that uses this type.
Purpose: To define a non-positive integer type.

2.1.9 NonNegativeNumber

Format: A *NonNegativeNumber*, a series of characters that together represent a non-negative (greater than or equal to zero) integer value, must conform to the following format:
NonNegativeNumber = "**0**" | *PositiveNumber*
Length: *NonNegativeNumbers* have a minimum length of one character; the maximum length is unspecified, but should be stated for each attribute that uses this type.
Purpose: To define a non-negative integer type.

2.1.10 Number

Format: A *Number*, a series of characters that together represent an integer value, must conform to the following format:
Number = *NegativeNumber* | "**0**" | *PositiveNumber*
Length: *Numbers* have a minimum length of one character; the maximum length is unspecified, but should be stated for each attribute that uses this type.
Purpose: To define an integer type.

2.1.11 ByteNum

Format: A *ByteNum*, representing an integer value that can be stored in one byte (eight bits), must conform to the following format:
ByteNum = [["**1**" | "**2**"] *Digit*] *Digit*
Length: *ByteNums* have a minimum length of one character and a maximum length of three characters.
Range: The minimum equivalent integral value for a *ByteNum* is 0 and the maximum is 255.
Purpose: To represent an integer value between 0 and 255 for use in other more complex types.

2.1.12 Hex

Format: A *Hex*, representing a single hexadecimal character, must conform to the following format:
Hex = *Digit* | "**A**" | "**B**" | "**C**" | "**D**" | "**E**" | "**F**" | "**a**" | "**b**" | "**c**" | "**d**" | "**e**" | "**f**"
Length: *Hexes* always have a length of one character.
Range: The minimum equivalent integral value for a *Hex* is 0 and the maximum is 15.
Purpose: To represent a single hexadecimal character for use in other more complex types.

2.1.13 Escaped

Format: An *Escaped*, an integer value that can be stored in one byte (eight bits) expressed in hexadecimal format, must conform to the following format:

Escaped = ("**&**" "**a**" "**m**" "**p**" "**;**") | ("**&**" "**l**" "**t**" "**;**") | ("**&**" "**g**" "**t**" "**;**") |
("**&**" "**a**" "**p**" "**o**" "**s**" "**;**") | ("**&**" "**q**" "**u**" "**o**" "**t**" "**;**") |
("**&**" "**#**" "**x**" *Hex Hex Hex Hex* "**;**")

Length: *Escaped* types have a minimum length of four characters and a maximum length of eight characters.

Range: The minimum equivalent integral value for an *Escaped* is 0 and the maximum is 255.

Purpose: To represent an integer value between 0 and 255 in hexadecimal-character format for use in other more complex types. It corresponds to the standard XML Encoding format, and as such the only valid values for the hexadecimal values represented by an Escaped character are the Horizontal Tab (x0009), Line Feed (x000A), Carriage Return (x000D), and the printable characters between x0020 and x007E (decimal 32 and 126), inclusive.

2.1.14 Mark

Format: A *Mark*, representing a single printable non-alphanumeric character, must conform to the following format:

Mark = "**-**" | "**_**" | "**.**" | "**!**" | "**~**" | "*****" | "**(**" | "**)**" | "\" | "/" | "**#**" | "**$**" | "**%**" | "**?**" | "**=**" | "**+**"

Length: *Marks* always have a length of one character.

Purpose: To represent a single printable non-alphanumeric character for use in other more complex types.

2.1.15 Unreserved

Format: An *Unreserved*, representing a single printable character that is available for common use, must conform to the following format:

Unreserved = *Mark* | *AlphaNum*

Length: *Unreserveds* always have a length of one character.

Purpose: To represent a single printable character that is available for common use in other more complex types.

2.1.16 PathChar

Format: A *PathChar*, representing a single character that is valid in directory paths, must conform to the following format:

PathChar = *Unreserved* | *Escaped*

Length: *PathChars* always have a length of one character.

Purpose: To represent a single character that is valid in directory paths.

2.1.17 WCTPChar

Format: A *WCTPChar*, representing a single character that is valid in most WCTP fields, must conform to the following format:
WCTPChar = *Unreserved* | *Escaped* | "**;**" | "**@**" | "**|**" | "**,**" | "**[**" | "**]**" | "**:**" | "**{**" | "**}**" | "^"
Length: *WCTPChars* always have a length of one character.
Purpose: To represent a single character that is valid in directory paths.

2.1.18 String

Format: A *String*, representing a string of characters that is valid in many WCTP fields, must conform to the following format:
String = 1**WCTPChar*
Length: *Strings* have a minimum length of one character; the maximum length is unspecified, but should be stated for each attribute that uses this type.
Purpose: To represent a character string that is valid in most WCTP attributes.

2.1.19 CapString

Format: A *CapString*, representing a string of digits or upper case letters, must conform to the following format:
CapString = 1*(*UpAlpha* | *Digit*)
Length: *CapStrings* have a minimum length of one character; the maximum length is unspecified, but should be stated for each attribute that uses this type.
Purpose: To represent an upper-case character string.

2.1.20 Version

Format: A *Version*, representing a string of characters that make up a valid WCTP version identifier, must conform to the following format:
Version = *CapString* "**-**" *CapString* "**-**" "**V**" 1**Digit* "**R**" 1**Digit*
Length: *Versions* have a minimum length of eight characters, and a maximum length of twenty-four characters.
Purpose: To represent a character string used for WCTP version identifiers.

2.1.21 PassCode

Format: A *PassCode*, representing a string of characters, must conform to the following format:
PassCode = 1**Unreserved*
Length: *PassCodes* have a minimum length of one character; the maximum length is unspecified, but should be stated for each attribute that uses this type.
Purpose: To represent a password character string.

2.1.22 PhoneNumber

Format: A *PhoneNumber*, representing a valid telephone number, must conform to the following format:
PhoneNumber = String
Length: *PhoneNumbers* have a minimum length of ten characters, and a maximum length of thirty-two characters.
Purpose: To represent a valid telephone number.

2.2 Enumerated Types

There are several enumerated types needed for fields in the WCTP. These enumerated types are defined here.

2.2.1 TrueFalse

Format: A *TrueFalse*, representing a case insensitive enumeration of character strings, must conform to the following format:
TrueFalse = ("**t**" "**r**" "**u**" "**e**") | ("**f**" "**a**" "**l**" "**s**" "**e**")
Length: *TrueFalses* have a length of either four or five characters.
Purpose: To represent the case-insensitive values 'true' and 'false' as the only valid options for a field.

2.2.2 YesNo

Format: A *YesNo*, representing a case insensitive enumeration of character strings, must conform to the following format:
YesNo = ("**y**" "**e**" "**s**") | ("**n**" "**o**")
Length: *YesNos* have a length of either two or three characters.
Purpose: To represent the case-insensitive values 'yes' and 'no' as the only valid options for a field.

2.2.3 DeliveryPriority

Format: A *DeliveryPriority*, representing a case insensitive enumeration of character strings, must conform to the following format:
DeliveryPriority = ("**H**" "**I**" "**G**" "**H**") | ("**N**" "**O**" "**R**" "**M**" "**A**" "**L**") | ("**L**" "**O**" "**W**")
Length: *DeliveryPrioritys* have a minimum length of three characters and a maximum length of six characters.
Purpose: To represent the case-insensitive values 'high', 'normal', and 'low' as the only valid options for a field.

2.2.4 Notification

Format: A *Notification*, representing a case insensitive enumeration of character strings, must conform to the following format:
Notification = ("**Q**" "**U**" "**E**" "**U**" "**E**" "**D**") | ("**D**" "**E**" "**L**" "**I**" "**V**" "**E**" "**R**" "**E**" "**D**") | ("**R**" "**E**" "**A**" "**D**")

Length: *Notifications* have a minimum length of four characters and a maximum length of nine characters.

Purpose: To represent the case-insensitive values 'queued', 'delivered', and 'read' as the only valid options for a field.

2.2.5 SupportType

Format: A *SupportType*, representing a case insensitive enumeration of character strings, must conform to the following format:
SupportType = ("**S**" "**u**" "**p**" "**p**" "**o**" "**r**" "**t**" "**e**" "**d**") | ("**D**" "**e**" "**p**" "**r**" "**e**" "**c**" "**a**" "**t**" "**e**" "**d**") | ("**N**" "**o**" "**t**" "**S**" "**u**" "**p**" "**p**" "**o**" "**r**" "**t**" "**e**" "**d**")

Length: *SupportTypes* have a minimum length of nine characters and a maximum length of twelve characters.

Purpose: To represent the case-insensitive values 'Supported', 'Deprecated', and 'NotSupported' as the only valid options for a field.

2.2.6 DataType

Format: A *DataType*, representing a case insensitive enumeration of character strings, must conform to the following format:
DataType = ("**O**" "**P**" "**A**" "**Q**" "**U**" "**E**") | ("**F**" "**L**" "**E**" "**X**" "**S**" "**u**" "**i**" "**t**" "**e**")

Length: *DataTypes* have a minimum length of six characters and a maximum length of nine characters.

Purpose: To represent the case-insensitive values 'OPAQUE' and 'FLEXSuite' as the only valid options for a field.

2.2.7 EncodingType

Format: An *EncodingType*, representing a case insensitive enumeration of character strings, must conform to the following format:
EncodingType = ("**s**" "**t**" "**a**" "**n**" "**d**" "**a**" "**r**" "**d**") | ("**b**" "**a**" "**s**" "**e**" "**6**" "**4**")

Length: *EncodingTypes* have a minimum length of six characters and a maximum length of eight characters.

Purpose: To represent the case-insensitive values 'standard' and 'base64' as the only valid options for a field.

2.2.8 FailReasonType

Format: A *FailReasonType*, representing a case insensitive enumeration of character strings, must conform to the following format:

FailReasonType = ("**u**" "**n**" "**k**" "**n**" "**o**" "**w**" "**n**") | ("**t**" "**e**" "**m**" "**p**" "**F**" "**a**" "**i**" "**l**" "**u**" "**r**" "**e**") | ("**i**" "**n**" "**v**" "**a**" "**l**" "**i**" "**d**" "**A**" "**u**" "**t**" "**h**" "**C**" "**o**" "**d**" "**e**")

Length: *FailReasonTypes* have a minimum length of seven characters and a maximum length of fifteen characters.

Purpose: To represent the case-insensitive values 'unknown', 'tempFailure', and 'invalidAuthCode' as the only valid options for a field.

2.3 Address Types

Section 2.4 of the WCTP 1.1 Protocol Specification defines an address by specifying the format for entity-addresses and transport addresses. It also provides examples of valid addresses for wireless devices and wireline hosts. The formats specified there for an address, entity-address, and transport-address are used as a base for the address types defined here.
All address types shown here must contain only characters that comply with RFC 2396, section 2, *URI Characters*.

2.3.1 Alias

Format: An *Alias* must conform to the following format:
Alias = *("!")*AlphaNum 1*Alpha *AlphaNum
Length: The minimum length is one character, and the maximum length is thirty-two characters.
Purpose: To define a shorter substitute for an address.

2.3.2 PollerID

Format: A *PollerID* must conform to the following format:
*PollerID = 1*Unreserved "." 1*Unreserved*
Length: The minimum length is three characters, and the maximum length is one hundred twenty-eight characters.
Purpose: To define an identifier for an enterprise organization to use wctp-PollForMessages.

2.3.3 Protocol

Format: A *Protocol* must conform to the following format:
*Protocol = Alpha *(Alpha | Digit* | "+" | "-" | ".")
Length: The minimum length is one character, and the maximum length is eight characters.
Purpose: To specify the protocol to be used in transporting a WCTP operation. Some common examples include "HTTP" and "FTP".

2.3.4 IPAddress

Format: An *IPAddress* must conform to the following format:
IPAddress = ByteNum "." *ByteNum* "." *ByteNum* "." *ByteNum*
Length: The minimum length is seven characters and the maximum length is fifteen characters.
Purpose: To specify the IP Address version of a domain name.

2.3.5 TopLabel

Format: A *TopLabel* must conform to the following format:
*TopLabel = Alpha | Alpha *(AlphaNum* | "-") *AlphaNum*
Length: The minimum length is one character and the maximum length is eight characters.
Purpose: To specify the top level portion or segment of a domain name.

2.3.6 DomainLabel

Format: A *DomainLabel* must conform to the following format:
*DomainLabel = AlphaNum | AlphaNum *(AlphaNum | "-") AlphaNum*
Length: The minimum length is one character and the maximum length is forty-eight characters.
Purpose: To specify a portion or segment of a domain name other than the top level.

2.3.7 HostName

Format: A *HostName* must conform to the following format:
HostName = 1*(*DomainLabel* "**.**") *TopLabel*
Length: The minimum length is three characters and the maximum length is one hundred twenty-eight characters. NOTE: HostName may be used as a component of TransportAddress and Address which are also constrained to one hundred twenty-eight characters. When used as a component of another field, the length of HostName must be limited such that the total field length is no greater than one hundered twenty-eight characters.
Purpose: To specify a non-IP address domain name for a computer.

2.3.8 Domain

Format: A *Domain* must conform to the following format:
Domain = HostName | IPAddress
Length: The minimum length is three characters and the maximum length is one hundred twenty-eight characters. NOTE: Domain may be used as a component of TransportAddress and Address which are also constrained to one hundred twenty-eight characters. When used as a component of another field, the length of Domain must be limited such that the total field length is no greater than one hundered twenty-eight characters.
Purpose: To specify an actual individual or application by name.

2.3.9 Port

Format: A *Port*, representing an integer, must conform to the following format:
Port = [[[["**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**"] *Digit*] *Digit*] *Digit*] *Digit*
Length: The minimum length is one character and the maximum length is five characters.
Range: The minimum numeric value is 1 and the maximum value is 65535.
Purpose: To specify a socket port involved in an address for a WCTP operation.

2.3.10 PathSegment

Format: A *PathSegment* must conform to the following format:
*PathSegment = 1*PathChar "/"*

Length: The minimum length is one character, and the maximum length is sixteen characters.
Purpose: To specify a segment of a directory path.

2.3.11 Path

Format: A *Path* must conform to the following format:
Path = "/" *(*PathSegment*) **PathChar*

Length: The minimum length is one character, and the maximum length is one hundred twenty-eight characters. NOTE: Path may be used as a component of TransportAddress, EntityAddress, and Address which are also constrained to one hundred twenty-eight characters. When used as a component of another field, the length of Path must be limited such that the total field length is no greater than one hundered twenty-eight characters.

Purpose: To specify an absolute path to an application to accept the operation.

2.3.12 TransportAddress

Format: A *TransportAddress* must conform to the following format:
TransportAddress = [*Protocol* "**:**" ["/" "/"]] *Domain* ["**:**" *Port*] [*Path*]

Length: The minimum length is three characters, and the maximum length is one hundred twenty-eight characters.

Purpose: To specify the full "transport protocol" portion of an address to be used in a WCTP operation. This generally represents a server on a network, including how to locate it and how to communicate with it.

2.3.13 Scheme

Format: A *Scheme* must conform to the following format:
Scheme = *Alpha* *(*Alpha* | *Digit* | "**+**" | "**-**" | "**.**")

Length: The minimum length is one character, and the maximum length is eight characters.

Purpose: To specify the communications scheme to be used in an address for a WCTP operation. Some common examples include "phoneto" and "faxto".

2.3.14 Entity

Format: An *Entity* must conform to the following format:
Entity = *1*WCTPChar*

Length: The minimum length is one character, and the maximum length is thirty-two characters.

Purpose: To specify an actual individual or application by name.

2.3.15 EntityAddress

Format: An *EntityAddress* must conform to the following format:
EntityAddress = [*Scheme* "**:**"] *Entity* ["**:**" *Port*][*Path*]

Length: The minimum length is one character, and the maximum length is one hundred twenty-eight characters.

Purpose: To specify the full "entity" portion of an address to be used in a WCTP operation. This generally represents an individual or application, including how to locate that entity.

2.3.16 Address

Format: An *Address* must conform to the following format:
Address = ([*EntityAddress* "**@**"] *TransportAddress*) | *Alias* | *PollerID*
Length: The minimum length is one character, and the maximum length is one hundred twenty-eight characters.
Purpose: To specify an address needed in a WCTP operation, such as a recipient or sender.

2.3.17 EmailAddress

Format: An *EmailAddress* in the WCTP must conform to the following format:
EmailAddress = *Entity* "**@**" *Domain*
Length: The minimum length is five characters, and the maximum length is one hundred twenty-eight characters.
Purpose: To specify a full SMTP-compliant email address to be used in WCTP operations.

2.3.18 WWWAddress

Format: A *WWWAddress* must conform to the following format:
WWWAddress = *Domain* [Path]
Length: The minimum length is three characters and the maximum length is one hundred twenty-eight characters.
Purpose: To specify a valid World-Wide-Web address to be used in a wctp-ContactInfo element. This type of address represents a web server that can be accessed by using a standard browser.

2.4 Date and Time Types

Section 3.2 of the WCTP 1.1 Protocol Specification defines the valid date, time, and datetime formats. It also provides examples of valid timestamps for WCTP operations. The formats specified there for a date, time and datetime stamp are used as a base for the date and time types defined here.

2.4.1 Day

Format: A *Day*, a two-digit value representing a calendar day, must conform to the following format:
Day = ("**0**" ("**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**" | "**7**" | "**8**" | "**9**")) | ("**1**" *Digit*) | ("**2**" *Digit*) |
("**3**" ("**0**" | "**1**"))
If the *Day* is associated with a *Month*, then the set of valid values may be further restricted to match the number of days in that month according to the Gregorian calendar, including taking into account leap years. For example, in a non-leap year such as 1999, the values "**2**" "**9**", "**3**" "**0**", and "**3**" "**1**" are NOT valid for the *Month* of "**0**" "**2**" (February).
Length: A *Day* must always have a length of two characters.
Purpose: To define a type that represents a valid day in a calendar month.

2.4.2 Month

Format: A *Month*, a two-digit value representing a calendar month, must conform to the following format:
Month = ("**0**" ("**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**" | "**7**" | "**8**" | "**9**")) | ("**1**" ("**0**" | "**1**" | "**2**"))
The value for January is "**0**" "**1**", for February "**0**" "**2**", and so on through December as "**1**" "**2**".
Length: A *Month* must always have a length of two characters.
Purpose: To define a type that represents a valid month in a calendar year.

2.4.3 Year

Format: A *Year*, a four-digit value representing a calendar year, must conform to the following format:
Year = Digit Digit Digit Digit
A *Year* has a set of valid values ranging from "**0**" "**0**" "**0**" "**0**" to "**9**" "**9**" "**9**" "**9**".
Length: A *Year* must always have a length of four characters.
Purpose: To define a type that represents a valid year in the Gregorian calendar.

2.4.4 Date

Format: A *Date*, a series of characters that represent a year, month, and day, must conform to the following format:
Date = *Year* "**-**" *Month* "**-**" *Day*
Length: A *Date* must always have a length of ten characters.
Purpose: To define a type that represents a valid date in the Gregorian calendar.

2.4.5 Seconds

Format: *Seconds*, a series of characters that represent the number of seconds past a particular minute, must conform to the following format:
Seconds = ("**0**" | "**1**" | "**2**" | "**3**" | "**4**" | "**5**") *Digit*
Seconds has a set of valid values ranging from "**0**" "**0**" to "**5**" "**9**".
Length: *Seconds* must always have a length of two characters.
Purpose: To define a type that represents a valid number of seconds past a minute.

2.4.6 Minutes

Format: *Minutes*, a series of characters that represent the number of minutes past a particular hour, must conform to the following format:
Minutes = ("**0**" | "**1**" | "**2**" | "**3**" | "**4**" | "**5**") *Digit*
Minutes has a set of valid values ranging from "**0**" "**0**" to "**5**" "**9**".
Length: *Minutes* must always have a length of two characters.
Purpose: To define a type that represents a valid number of minutes past an hour.

2.4.7 Hours

Format: *Hours*, a series of characters that represent the number of hours past midnight on a particular day, must conform to the following format:
Hours = ("**0**" *Digit*) | ("**1**" *Digit*) | ("**2**" ("**0**" | "**1**" | "**2**" | "**3**"))
Hours has a set of valid values ranging from "**0**" "**0**" to "**2**" "**3**".
Length: *Hours* must always have a length of two characters.
Purpose: To define a type that represents a valid number of hours past midnight in a particular day.

2.4.8 Time

Format: A *Time*, a series of characters that represent the number of hours, minutes, and seconds (along with an optional subseconds value) past midnight on a particular day, must conform to the following format:
Time = ("**2**" "**4**" "**:**" "**0**" "**0**" "**:**" "**0**" "**0**") | (*Hours* "**:**" *Minutes* "**:**" *Seconds* ["**,**" *Digit* [*Digit* [*Digit*]]])
Time includes the value of '24:00:00', which is equivalent to '00:00:00', and both of which represent midnight, but potentially on the boundary between different dates. For example, '24:00:00' on June 3rd is NOT equivalent to '00:00:00' on June 3rd, but rather is equivalent to '00:00:00' on June 4th.
Length: A *Time* has a minimum length of eight characters and a maximum length of twelve characters.
Purpose: To define a type that represents a valid time during a day.

2.4.9 DateTime

Format: A *DateTime*, a series of characters that represent a year, month, and day, together with a time expressed as hours, minutes, and seconds, must conform to the following format:
DateTime = Date "**T**" *Time*
Length: A *DateTime* has a minimum length of nineteen characters and a maximum length of twenty-three characters.
Purpose: To define a type that represents a valid date and time in the Gregorian calendar.

2.4.10 TimeInterval

Format: A *TimeInterval*, a series of characters representing an integer as a period of elapsed time in seconds, must conform to the following format:
TimeInterval = NonNegativeNumber
Length: A *TimeInterval* has a minimum length of one character; the maximum length is unspecified, but should be stated for each attribute that uses this type.
Range: The minimum numeric value is 0; the maximum numeric value should be stated for each attribute that uses this type.
Purpose: To define a type that represents a valid time interval in seconds.

3 Type Summary

This document uses a BNF-like grammar as defined in RFC2396. Many of the details presented here are similar to those defined in RFC2396, although some have been changed to meet the needs of the WCTP.

Briefly, rules are separated from definitions by an equal "=", indentation is used to continue a rule definition over more than one line, literals are quoted with "", parentheses "(" and ")" are used to group elements, optional elements are enclosed in "[" and "]" brackets, "|" is used to designate alternatives, and elements may be preceded with n* to designate n or more repetitions of the following element, where n is optional and defaults to 0.

Unlike many specifications that use a BNF-like grammar to define the bytes (octets) allowed by a protocol, this grammar is defined in terms of characters. Each literal in the grammar corresponds to the character it represents, rather than to the octet encoding of that character in any particular coded character set. How WCTP operations are represented in terms of bits and bytes on the wire is dependent upon the character encoding of the protocol used to transport it, or the charset of the document that contains it.

3.1 Basic Types

The following basic field types are used in multiple field format definitions.

```
Digit             =  "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
                     "8" | "9"

LowAlpha          =  "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" |
                     "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" |
                     "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" |
                     "y" | "z"

UpAlpha           =  "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" |
                     "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" |
                     "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" |
                     "Y" | "Z"

Alpha             =  LowAlpha | UpAlpha

AlphaNum          =  Alpha | Digit

NegativeNumber    =  "-" ( "1" | "2" | "3" | "4" | "5" | "6" | "7" |
                     "8" | "9" ) *Digit

PositiveNumber    =  ( "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" |
                     "9" ) *Digit

NonPositiveNumber =  NegativeNumber | "0"

NonNegativeNumber =  "0" | PositiveNumber

Number            =  NegativeNumber | "0" | PositiveNumber

ByteNum           =  [ [ "1" | "2" ] Digit ] Digit

Hex               =  Digit | "A" | "B" | "C" | "D" | "E" | "F" | "a" |
                     "b" | "c" | "d" | "e" | "f"

Escaped           =  ( "&" "a" "m" "p" ";" ) | ( "&" "l" "t" ";" ) | (
                     "&" "g" "t" ";" ) | ( "&" "a" "p" "o" "s" ";" ) |
                     ( "&" "q" "u" "o" "t" ";" ) |( "&" "#" "x" Hex Hex
                     Hex Hex ";" )

Mark              =  "-" | "_" | "." | "!" | "~" | "*" | "(" | ")" |
                     "\" | "/" | "#" | "$" | "%" | "?" | "=" | "+"

Unreserved        =  Mark | AlphaNum

PathChar          =  Unreserved | Escaped

WCTPChar          =  Unreserved | Escaped | ";" | "@" | "|" | "," | "["
                     | "]" | ":" | "{" | "}" | "^"

String            =  1*WCTPChar

CapString         =  1*( UpAlpha | Digit )

Version           =  CapString "-" CapString "-" "V" 1*Digit "R"
                     1*Digit

PassCode          =  1*Unreserved

PhoneNumber       =  String
```

3.2 Enumerated Types

The following complex types represent the enumerated field types used in the WCTP field format definitions.

```
TrueFalse         =  ( "t" "r" "u" "e" ) | ( "f" "a" "l" "s" "e" )

YesNo             =  ( "y" "e" "s" ) | ( "n" "o")

DeliveryPriority  =  ( "H" "I" "G" "H" ) | ( "N" "O" "R" "M" "A" "L" )
                     | ( "L" "O" "W" )
Notification      =  ( "Q" "U" "E" "U" "E" "D" ) | ( "D" "E" "L" "I"
                     "V" "E" "R" "E" "D" ) | ( "R" "E" "A" "D" )
SupportType       =  ( "S" "u" "p" "p" "o" "r" "t" "e" "d" ) | ( "D"
                     "e" "p" "r" "e" "c" "a" "t" "e" "d" ) | ( "N" "o"
                     "t" "S" "u" "p" "p" "o" "r" "t" "e" "d" )
DataType          =  ( "O" "P" "A" "Q" "U" "E" ) | ( "F" "L" "E" "X"
                     "S" "u" "i" "t" "e" )
EncodingType      =  ( "s" "t" "a" "n" "d" "a" "r" "d" ) | ( "b" "a"
                     "s" "e" "6" "4" )
FailReasonType    =  ( "u" "n" "k" "n" "o" "w" "n" ) | ( "t" "e" "m"
                     "p" "F" "a" "i" "l" "u" "r" "e" ) | ( "i" "n" "v"
                     "a" "l" "i" "d" "A" "u" "t" "h" "C" "o" "d" "e" )
```

3.3 Address Types

The following complex types represent the more detailed and complicated field types used in the WCTP address field format definitions.

```
Alias              =   1*Unreserved
PollerID           =   1*Unreserved "." 1*Unreserved
Protocol           =   Alpha *( Alpha | Digit | "+" | "-" | "." )
IPAddress          =   ByteNum "." ByteNum "." ByteNum "." ByteNum
TopLabel           =   Alpha | Alpha *( AlphaNum | "-" ) AlphaNum
DomainLabel        =   AlphaNum | AlphaNum *( AlphaNum | "-" ) AlphaNum
HostName           =   1*( DomainLabel "." ) TopLabel
Domain             =   HostName | IPAddress
Port               =   [ [ [ [ "1" | "2" | "3" | "4" | "5" | "6" ] Digit
                       ] Digit ] Digit ] Digit
PathSegment        =   1*PathChar "/"
Path               =   "/" *( PathSegment ) *PathChar
TransportAddress       = [ Protocol ":" [ "/" "/" ] ] Domain [ ":" Port ]
                         [ Path ]
Scheme             =   Alpha *( Alpha | Digit | "+" | "-" | "." )
Entity             =   1*WCTPChar
EntityAddress      = [ Scheme ":" ] Entity [":" Port ][ Path ]
Address            = [ EntityAddress "@" ] TransportAddress
EmailAddress       = Entity "@" Domain
WWWAddress         = Domain [ Path ]
```

3.4 Date and Time Types

The following complex types represent the more detailed and complicated field types used in the WCTP date and time field format definitions.

```
Day             =  ( "0" ( "1" | "2" | "3" | "4" | "5" | "6" | "7" |
                   "8" | "9" ) ) | ( "1" Digit ) | ( "2" Digit ) | (
                   "3" ( "0" | "1" ) )

Month           =  ( "0" ( "1" | "2" | "3" | "4" | "5" | "6" | "7" |
                   "8" | "9" ) ) | ( "1" ( "0" | "1" | "2" ) )

Year            =  Digit Digit Digit Digit

Date            =  Year "-" Month "-" Day

Seconds         =  ( "0" | "1" | "2" | "3" | "4" | "5" ) Digit

Minutes         =  ( "0" | "1" | "2" | "3" | "4" | "5" ) Digit

Hours           =  ( "0" Digit ) | ( "1" Digit ) | ( "2" ( "0" | "1"
                   | "2" | "3" ) )

Time            =  ( "2" "4" ":" "0" "0" ":" "0" "0" ) | ( Hours ":"
                   Minutes ":" Seconds [ "," Digit [ Digit [ Digit ]
                   ] ] )

DateTime        =  Date "T" Time

TimeInterval    =  1*Digit
```

4 Attribute Definitions

Each attribute in the WCTP protocol is described in the below sections. Whenever possible, a field is described using one of the data types defined earlier.
The column titled "Attribute Name" gives the XML attribute name from the WCTP DTD.
The column titled "Disposition" refers to whether the attribute is Optional, Mandatory (Required), or Conditional.
The column titled "Type" gives the date type on which this attribute is based.
The column titled "Range" refers to the valid range of numeric values for the attribute. This column is only applicable when the attribute represents a numeric value. Otherwise, it contains "N/A" for "Not Applicable."
The column titled "Length" defines the minimum and maximum number of characters that can be placed in the attribute whenever it is present.
The column titled "Default" defines the default value for the attribute (if any). This column is only applicable when the attribute represents an attribute with an optional value in the DTD. Otherwise, it contains "N/A" for "Not Applicable."

4.1 wctp-Operation

Attribute Name	Disposition	Type	Range	Length	Default
wctpVersion	Mandatory	Version	N/A	8 to 24	N/A
wctpToken	Optional	String	N/A	1 to 16	N/A

4.2 wctp-SubmitHeader

Attribute Name	Disposition	Type	Range	Length	Default
submitTimestamp	Mandatory	DateTime	N/A	19 to 23	N/A

4.3 wctp-Originator

Attribute Name	Disposition	Type	Range	Length	Default
senderID	Mandatory	Address	N/A	1 to 128	N/A
securityCode	Optional	PassCode	N/A	1 to 16	N/A
miscInfo	Optional	String	N/A	1 to 16	N/A

4.4 wctp-MessageControl

Attribute Name	Disposition	Type	Range	Length	Default
messageID	Mandatory	String	N/A	1 to 32	N/A
transactionID	Optional	String	N/A	1 to 32	N/A
sendResponsesToID	Optional	Address	N/A	1 to 128	N/A
allowResponse	Optional	TrueFalse	N/A	4 to 5	true
notifyWhenQueued	Optional	TrueFalse	N/A	4 to 5	false
notifyWhenDelivered	Optional	TrueFalse	N/A	4 to 5	false
notifyWhenRead	Optional	TrueFalse	N/A	4 to 5	false
deliveryPriority	Optional	DeliveryPriority	N/A	3 to 6	NORMAL
deliveryBefore	Optional	DateTime	N/A	19 to 23	N/A
deliveryAfter	Optional	DateTime	N/A	19 to 23	N/A
preformatted	Optional	DateTime	N/A	4 to 5	false
allowTruncation	Optional	DateTime	N/A	4 to 5	true

4.5 wctp-Recipient

Attribute Name	Disposition	Type	Range	Length	Default
recipientID	Mandatory	Address	N/A	1 to 128	N/A
authorizationCode	Optional	PassCode	N/A	1 to 16	N/A

4.6 wctp-Alphanumeric

Attribute Name	Disposition	Type	Range	Length	Default
(PCDATA)	Optional	String	N/A	1 to 65535	N/A

4.7 wctp-TransparentData

Attribute Name	Disposition	Type	Range	Length	Default
(PCDATA)	Optional	String	N/A	1 to 65535	N/A
type	Optional	DataType	N/A	6 to 9	OPAQUE
encoding	Optional	EncodingType	N/A	6 to 8	base64

4.8 wctp-MessageText

Attribute Name	Disposition	Type	Range	Length	Default
(PCDATA)	Optional	String	N/A	1 to 65535	N/A

4.9 wctp-Choice

Attribute Name	Disposition	Type	Range	Length	Default
(PCDATA)	Mandatory	String	N/A	1 to 32	N/A

4.10 wctp-ResponseHeader

Attribute Name	Disposition	Type	Range	Length	Default
responseToMessageID	Mandatory	String	N/A	1 to 32	N/A
responseTimestamp	Optional	DateTime	N.A	19 to 23	N/A
respondingToTimestamp	Optional	DateTime	N/A	19 to 23	N/A
onBehalfOfRecipientID	Optional	Address	N.A	1 to 128	N/A

4.11 wctp-Notification

Attribute Name	Disposition	Type	Range	Length	Default
type	Mandatory	Notification	N/A	4 to 9	N/A

4.12 wctp-Failure

Attribute Name	Disposition	Type	Range	Length	Default
(PCDATA)	Optional	String	N/A	1 to 254	N/A
errorCode	Mandatory	PositiveNumber	1 to 9999	1 to 4	N/A
errorText	Optional	String	N/A	1 to 128	N/A

4.13 wctp-Success

Attribute Name	Disposition	Type	Range	Length	Default
(PCDATA)	Optional	String	N/A	1 to 254	N/A
successCode	Mandatory	PositiveNumber	1 to 9999	1 to 4	N/A
successText	Optional	String	N/A	1 to 128	N/A

4.14 wctp-PollForMessages

Attribute Name	Disposition	Type	Range	Length	Default
pollerID	Mandatory	PollerID	N/A	1 to 128	N/A
securityCode	Optional	PassCode	N/A	1 to 16	N/A
maxMessagesInBatch	Optional	NonNegative Number	0 to 99	1 to 2	N/A

4.15 wctp-PollResponse

Attribute Name	Disposition	Type	Range	Length	Default
minNextPollInterval	Optional	TimeInterval	0 to 3600	1 to 4	N/A

4.16 wctp-MessageReceived

Attribute Name	Disposition	Type	Range	Length	Default
sequenceNo	Mandatory	String	N/A	1 to 32	N/A

4.17 wctp-Message

Attribute Name	Disposition	Type	Range	Length	Default
sequenceNo	Mandatory	String	N/A	1 to 32	N/A

4.18 wctp-SubmitClientHeader

Attribute Name	Disposition	Type	Range	Length	Default
submitTimestamp	Mandatory	DateTime	N/A	19 to 23	N/A

4.19 wctp-ClientOriginator

Attribute Name	Disposition	Type	Range	Length	Default
senderID	Mandatory	Address	N/A	1 to 128	N/A
miscInfo	Optional	String	N/A	1 to 16	N/A

4.20 wctp-ClientMessageControl

Attribute Name	Disposition	Type	Range	Length	Default
sendResponsesToID	Mandatory	Address	N/A	1 to 128	N/A
allowResponse	Optional	TrueFalse	N/A	4 to 5	true
notifyWhenQueued	Optional	TrueFalse	N/A	4 to 5	false
notifyWhenDelivered	Optional	TrueFalse	N/A	4 to 5	false
notifyWhenRead	Optional	TrueFalse	N/A	4 to 5	false
deliveryPriority	Optional	DeliveryPriority	N/A	3 to 6	NORMAL
deliveryBefore	Optional	DateTime	N/A	19 to 23	N/A
deliveryAfter	Optional	DateTime	N/A	19 to 23	N/A
preformatted	Optional	DateTime	N/A	4 to 5	false
allowTruncation	Optional	DateTime	N/A	4 to 5	true

4.21 wctp-ClientSuccess

Attribute Name	Disposition	Type	Range	Length	Default
(PCDATA)	Optional	String	N/A	1 to 254	N/A
successCode	Mandatory	PositiveNumber	1 to 9999	1 to 4	N/A
successText	Optional	String	N/A	1 to 128	N/A
trackingNumber	Mandatory	String	N/A	1 to 16	N/A

4.22 wctp-ClientQuery

Attribute Name	Disposition	Type	Range	Length	Default
senderID	Mandatory	Address	N/A	1 to 128	N/A
recipientID	Mandatory	Address	N/A	1 to 128	N/A
trackingNumber	Mandatory	String	N/A	1 to 16	N/A

4.23 wctp-ClientQueryResponse

Attribute Name	Disposition	Type	Range	Length	Default
minNextPollInterval	Optional	String	0 to 3600	1 to 4	N/A

4.24 wctp-ClientResponseHeader

Attribute Name	Disposition	Type	Range	Length	Default
responseTimestamp	Optional	DateTime	N/A	19 to 23	N/A
respondingToTimestamp	Optional	DateTime	N/A	19 to 23	N/A

4.25 wctp-VersionQuery

Attribute Name	Disposition	Type	Range	Length	Default
inquirer	Mandatory	String	N/A	1 to 128	N/A
dateTime	Optional	DateTime	N/A	19 to 23	N/A
listDTDs	Optional	YesNo	N/A	2 to 3	N/A
listConfiguration	Option	YesNo	N/A	2 to 3	N/A

4.26 wctp-VersionResponse

Attribute Name	Disposition	Type	Range	Length	Default
responder	Mandatory	TransportAddress	N/A	3 to 128	N/A
dateTimeOfRep	Optional	DateTime	N/A	19 to 23	N/A
inquirer	Conditional[1]	String	N/A	1 to 128	N/A
dateTimeOfReq	Conditional[2]	DateTime	N/A	19 to 23	N/A
invalidAfter	Optional	DateTime	N/A	19 to 23	N/A
listDTDs	Optional	YesNo	N/A	2 to 3	N/A
listConfiguration	Optional	YesNo	N/A	2 to 3	N/A

4.27 wctp-ContactInfo

Attribute Name	Disposition	Type	Range	Length	Default
email	Optional	EmailAddress	N/A	5 to 128	N/A
phone	Optional	PhoneNumber	N/A	10 to 32	N/A
www	Optional	WWWAddress	N/A	3 to 128	N/A
info	Optional	String	N/A	1 to 255	N/A

4.28 wctp-DTDSupport

Attribute Name	Disposition	Type	Range	Length	Default
dtdName	Mandatory	Version	N/A	8 to 24	N/A
verToken	Optional	String	N.A	1 to 16	N/A
supportType	Optional	SupportType	N/A	9 to 12	Supported
exceptions	Optional	YesNo	N/A	2 to 3	N/A
supportUntil	Optional	DateTime	N/A	19 to 23	N/A
replacement	Optional	Version	N/A	8 to 24	N/A

[1] Only returned if provided in original request.

[2] Only returned if provided in original request.

4.29 wctp-MessageText

Attribute Name	Disposition	Type	Range	Length	Default
(PCDATA)	Optional	String	N/A	1 to 65535	N/A

4.30 wctp-LookupMessageControl

Attribute Name	Disposition	Type	Range	Length	Default
messageID	Mandatory	String	N/A	1 to 32	N/A
transactionID	Optional	String	N/A	1 to 32	N/A
sendResponsesToID	Optional	Address	N/A	1 to 128	N/A

4.31 wctp-LookupRespons

Attribute Name	Disposition	Type	Range	Length	Default
responseToMessageID	Mandatory	String	N/A	1 to 32	N/A
transactionID	Optional	String	N/A	1 to 32	

4.32 wctp-LookupData

Attribute Name	Disposition	Type	Range	Length	Default
maxMessageLength	Mandatory	NonNegative Number	0 to 10,485,760	1 to 8	N/A
mcrSupported	Optional	TrueFalse	N/A	4 to 5	N/A
canRespond	Optional	TrueFalse	N/A	4 to 5	N/A

4.33 wctp-ReturnToSvc

Attribute Name	Disposition	Type	Range	Length	Default
(PCDATA)	Optional	Address	N/A	1 to 128	N/A

4.34 wctp-MsgMultiHeader

Attribute Name	Disposition	Type	Range	Length	Default
submitTimestamp	Mandatory	DateTime	N/A	19 to 23	N/A

4.35 wctp-MsgMultiControl

Attribute Name	Disposition	Type	Range	Length	Default
messageID	Mandatory	String	N/A	1 to 32	N/A
transactionID	Optional	String	N/A	1 to 32	N/A
sendResponsesToID	Optional	Address	N/A	1 to 128	N/A
allowResponse	Optional	TrueFalse	N/A	4 to 5	true
notifyWhenQueued	Optional	TrueFalse	N/A	4 to 5	false
notifyWhenDelivered	Optional	TrueFalse	N/A	4 to 5	false
notifyWhenRead	Optional	TrueFalse	N/A	4 to 5	false
deliveryPriority	Optional	DeliveryPriority	N/A	3 to 6	NORMAL
deliveryBefore	Optional	DateTime	N/A	19 to 23	N/A
deliveryAfter	Optional	DateTime	N/A	19 to 23	N/A
preformatted	Optional	DateTime	N/A	4 to 5	false
allowTruncation	Optional	DateTime	N/A	4 to 5	true
allRecipsRequired	Optional	TrueFalse	N/A	4 to 5	true

4.36 wctp-SendMsgMultiResponse

Attribute Name	Disposition	Type	Range	Length	Default
maxNumRecips	Mandatory	PositiveNumber	1 to 9999	1 to 4	N/A
numValidRecips	Optional	NonNegative Number	0 to 9999	1 to 4	N/A
numInvalidRecips	Optional	NonNegative Number	0 to 9999	1 to 4	N/A

4.37 wctp-FailedRecipient

Attribute Name	Disposition	Type	Range	Length	Default
recipientId	Mandatory	Address	N/A	1 to 128	N/A
errorCode	Mandatory	PositiveNumber	1 to 9999	1 to 4	N/A
errorText	Optional	String	N/A	1 to 128	N/A