

Specification for a Lightweight Microbrowser for interactive tv applications, based on and compatible with WML

European Broadcasting Union



Union Européenne de Radio-Télévision

EBU·UER



Reference

DTS/JTC-016

Keywords

broadcasting, content, digital, TV

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, send your comment to:

editor@etsi.org

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2004.

© European Broadcasting Union 2004.

All rights reserved.

DECT™, **PLUGTESTS™** and **UMTS™** are Trade Marks of ETSI registered for the benefit of its Members.
TIPHON™ and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Contents

Intellectual Property Rights	12
Foreword.....	12
Introduction	12
1 Scope	14
2 References	15
3 Definitions, abbreviations and typographical conventions	15
3.1 Definitions	15
3.2 Abbreviations	16
3.3 Typographical conventions	16
4 WTVML XSDs	17
4.1 Introduction	17
5 WTVML specification	19
5.1 Decks and cards.....	19
5.1.1 WML element.....	19
5.1.1.1 Mode attribute	20
5.1.1.2 Update attribute.....	20
5.1.1.3 Attributes defined elsewhere.....	20
5.1.1.4 Example	20
5.1.2 Precache element	21
5.1.2.1 Href attribute	21
5.1.2.2 Cache-set attribute.....	21
5.1.2.3 The cache-control attribute.....	22
5.1.3 Void	22
5.1.4 Card element.....	23
5.1.4.1 Mode attribute	24
5.1.4.2 Paragap attribute	24
5.1.4.3 Scroll attribute.....	24
5.1.4.4 Tabindex attribute	24
5.1.4.5 Hint attribute	24
5.1.4.6 Hspace attribute.....	24
5.1.4.7 Vspace attribute.....	24
5.1.4.8 Navigation attribute.....	25
5.1.4.9 Example	25
5.2 Event bindings.....	26
5.2.1 Onevent element	26
5.2.1.1 Type attribute	27
5.2.2 Timer element.....	27
5.2.2.1 Value attribute.....	27
5.2.2.2 Retro attribute	28
5.2.3 Do element.....	29
5.2.3.1 Type attribute	30
5.2.3.2 Label attribute	30
5.2.3.3 Name attribute.....	30
5.2.3.4 Title attribute.....	30
5.2.3.5 Focus attribute.....	31
5.2.3.6 Void.....	31
5.2.3.7 Localsrc attribute.....	31
5.2.3.8 Icon attribute	31
5.2.3.9 Font attribute	31
5.2.3.10 Bglobalsrc attribute	31
5.2.3.11 Bgoffset attribute.....	31
5.2.3.12 Pos attribute.....	32
5.2.3.13 Example	32

5.3	Tasks	33
5.3.1	Go element	33
5.3.1.1	Setattr subelement	34
5.3.1.2	Sendreferer attribute	34
5.3.1.3	Target attribute	34
5.3.1.4	Cache-control attribute	34
5.3.1.5	Timeout attribute	34
5.3.1.6	Example	34
5.3.2	Prev element	36
5.3.2.1	Settattr subelement	36
5.3.2.2	Example	36
5.3.3	Refresh element	37
5.3.3.1	Settattr subelement	37
5.3.4	Noop element	38
5.3.4.1	Settattr subelement	38
5.3.4.2	Example	38
5.4	Variables	39
5.4.1	Setvar element	39
5.4.1.1	Name attribute	40
5.4.1.2	Cache-control attribute	40
5.4.2	Setattr element	40
5.4.2.1	Name attribute	41
5.4.2.2	Value attribute	41
5.4.2.3	Examples	41
5.4.3	Variable retrieval and setattr syntax	41
5.5	Card fields	43
5.5.1	Select element	43
5.5.1.1	Localsrc attribute	44
5.5.1.2	Style attribute	44
5.5.1.3	Pivot attribute	44
5.5.2	Optgroup element	45
5.5.2.1	Localsrc attribute	45
5.5.3	Option element	46
5.5.3.1	Localsrc attribute	46
5.5.4	Input element	47
5.5.4.1	Name attribute	48
5.5.4.2	Type attribute	48
5.5.4.3	Value attribute	48
5.5.4.4	Format attribute	48
5.5.4.5	Size attribute	49
5.5.4.6	Maxlength attribute	49
5.5.4.7	Emptyok attribute	50
5.5.4.8	Borderwidth attribute	50
5.5.4.9	Bgcolor attribute	50
5.5.4.10	Cursorcolor attribute	50
5.5.4.11	Bordercolor attribute	50
5.5.4.12	Borderunfocused attribute	50
5.5.4.13	Borderfocused attribute	50
5.5.4.14	Label attribute	50
5.5.4.15	Localsrc attribute	50
5.5.4.16	Pos attribute	51
5.5.4.17	Example	51
5.6	Images	52
5.6.1	Img element	52
5.6.1.1	Frames attribute	53
5.6.1.2	Loop attribute	53
5.6.1.3	Ticks attribute	53
5.6.1.4	Index attribute	53
5.6.1.5	Cache-control attribute	53
5.7	Tables	54
5.7.1	Table element	54
5.7.1.1	Align attribute	55

5.7.1.2	Columns attribute	55
5.7.1.3	Rowgap attribute	55
5.7.1.4	Colgap attribute	55
5.7.1.5	Vspace attribute.....	55
5.7.1.6	Hspace attribute.....	55
5.7.1.7	Hpad attribute.....	55
5.7.1.8	Vpad attribute.....	55
5.7.1.9	Colwidths attribute.....	56
5.7.1.10	Bgimage attribute.....	56
5.7.1.11	Bgcolor attribute	56
5.7.1.12	Bgradius attribute.....	56
5.7.1.13	Bordercolor attribute	56
5.7.1.14	Borderwidth attribute	56
5.7.1.15	Update attribute.....	56
5.7.1.16	Example	57
5.7.2	Tr element.....	58
5.7.2.1	Title attribute.....	59
5.7.2.2	Height attribute	59
5.7.2.3	Localsrc attribute.....	59
5.7.2.4	Bgimage attribute.....	59
5.7.2.5	Bgradius attribute.....	59
5.7.2.6	Bgcolor attributes.....	59
5.7.2.7	Borderwidth attribute	59
5.7.2.8	Bordercolor attributes.....	59
5.7.2.9	Update attribute.....	60
5.7.2.10	Example	60
5.7.3	Td element	61
5.7.3.1	Align attribute	63
5.7.3.2	Title attribute.....	63
5.7.3.3	Height attribute	63
5.7.3.4	Width attribute	63
5.7.3.5	Hspace attribute.....	63
5.7.3.6	Vspace attribute.....	63
5.7.3.7	Hpad attribute.....	64
5.7.3.8	Vpad attribute.....	64
5.7.3.9	Linegap attribute	64
5.7.3.10	Colspan attribute	64
5.7.3.11	Rowspan attribute	64
5.7.3.12	Localsrc attribute.....	64
5.7.3.13	Bgimage attribute.....	64
5.7.3.14	Bgradius attribute.....	64
5.7.3.15	Bgcolor attributes.....	64
5.7.3.16	Borderwidth attribute	65
5.7.3.17	Bordercolor attributes.....	65
5.7.3.18	Mode attribute	65
5.7.3.19	Update attribute.....	65
5.7.3.20	Example	65
5.8	Text	67
5.8.1	Font element	67
5.8.1.1	Size attribute	68
5.8.1.2	Fgstatic attribute.....	68
5.8.1.3	Fgcolor attribute.....	68
5.8.1.4	Fgfocused attribute.....	68
5.8.1.5	Fgactivated attribute.....	68
5.8.1.6	Example	68
5.8.2	Paragraphs.....	69
5.8.2.1	Linegap attribute	71
5.8.2.2	Update attribute.....	71
5.8.2.3	Example	71
6	WTVML 6.3 Modeset specification.....	72
6.1	Introduction	72

6.2	Cache control.....	72
6.2.1	Head element	72
6.2.2	Meta element	72
6.2.2.1	Http-equiv attribute	72
6.2.2.2	Content attribute.....	73
6.3	WTVML 6.3 Screen objects.....	73
6.3.1	AV element.....	73
6.3.1.1	Id attribute.....	74
6.3.1.2	Mode attribute.....	74
6.3.1.3	Rect-src attribute.....	74
6.3.1.4	Rect attribute.....	75
6.3.1.5	Hscale attribute.....	75
6.3.1.6	Vscale attribute.....	75
6.3.1.7	Video attribute.....	75
6.3.1.8	Audio attribute.....	76
6.3.1.9	Toggle attribute.....	76
6.3.1.10	Offset attribute.....	76
6.3.1.11	Bgimage attribute.....	76
6.3.1.12	Service attribute.....	77
6.3.1.13	Example.....	77
6.3.2	Status element.....	77
6.3.2.1	Id attribute.....	78
6.3.2.2	Align attribute.....	78
6.3.2.3	Rect attribute.....	78
6.3.2.4	Examples.....	78
6.3.3	Resume element.....	79
6.3.3.1	Id attribute.....	79
6.3.3.2	Localsrc attribute.....	79
6.3.3.3	Rect attribute.....	79
6.3.3.4	Example.....	79
6.3.4	Img element.....	80
6.3.4.1	Id attribute.....	80
6.3.4.2	Src attribute.....	81
6.3.4.3	Localsrc attribute.....	81
6.3.4.4	Vspace attribute.....	81
6.3.4.5	Hspace attribute.....	81
6.3.4.6	Align attribute.....	81
6.3.4.7	Height & width attributes.....	81
6.3.4.8	Rect attribute.....	81
6.3.4.9	Visible attribute.....	81
6.3.4.10	Cache control attribute.....	82
6.3.4.11	Example.....	82
6.3.5	Clock element.....	82
6.3.5.1	Id attribute.....	83
6.3.5.2	Format attribute.....	83
6.3.5.3	Value attribute.....	83
6.3.5.4	Visible attribute.....	84
6.3.5.5	Font attribute.....	84
6.3.5.6	Rect attribute.....	84
6.3.5.7	Align attribute.....	84
6.3.5.8	%Wcell attributes.....	84
6.3.5.9	Example.....	84
6.3.6	Title element.....	85
6.3.6.1	Id attribute.....	85
6.3.6.2	Value attribute.....	86
6.3.6.3	Visible attribute.....	86
6.3.6.4	Font attribute.....	86
6.3.6.5	Rect attribute.....	86
6.3.6.6	Align attribute.....	86
6.3.6.7	Hpad attribute.....	86
6.3.6.8	Vpad attribute.....	86
6.3.6.9	Aacolor attribute.....	86

6.3.6.10	Bgcolor attribute	86
6.3.6.11	Bgradient attribute	87
6.3.6.12	Bordercolor attribute	87
6.3.6.13	Borderwidth attribute	87
6.3.6.14	Example	87
6.3.7	Wtvm:precache element	87
6.3.7.1	Href element	88
6.3.7.2	Cache-set attribute	88
6.3.8	Cache-control attribute	88
6.3.9	Hint element	89
6.3.9.1	Id attribute	90
6.3.9.2	Value attribute	90
6.3.9.3	Visible attribute	90
6.3.9.4	Font attribute	90
6.3.9.5	Rect attribute	90
6.3.9.6	Align attribute	90
6.3.9.7	Hpad attribute	90
6.3.9.8	Vpad attribute	90
6.3.9.9	Aacolor attribute	90
6.3.9.10	Bgcolor attribute	91
6.3.9.11	Bgradient attribute	91
6.3.9.12	Bordercolor attribute	91
6.3.9.13	Borderwidth attribute	91
6.3.9.14	Example	91
6.3.10	Callrate element	92
6.3.10.1	Id attribute	93
6.3.10.2	Value attribute	93
6.3.10.3	Visible attribute	93
6.3.10.4	Font attribute	93
6.3.10.5	Rect attribute	93
6.3.10.6	Align attribute	93
6.3.10.7	Hpad attribute	93
6.3.10.8	Vpad attribute	93
6.3.10.9	Aacolor attribute	93
6.3.10.10	Bgcolor attribute	94
6.3.10.11	Bgradient attribute	94
6.3.10.12	Bordercolor attribute	94
6.3.10.13	Borderwidth attribute	94
6.3.10.14	Example	94
6.3.11	Text element	95
6.3.11.1	Id attribute	96
6.3.11.2	Value attribute	96
6.3.11.3	Visible attribute	96
6.3.11.4	Font attribute	96
6.3.11.5	Rect attribute	96
6.3.11.6	Align attribute	96
6.3.11.7	Hpad attribute	96
6.3.11.8	Vpad attribute	96
6.3.11.9	Aacolor attribute	97
6.3.11.10	Bgcolor attribute	97
6.3.11.11	Bgradient attribute	97
6.3.11.12	Bordercolor attribute	97
6.3.11.13	Borderwidth attribute	97
6.3.11.14	Example	97
6.3.12	Box element	98
6.3.12.1	Id attribute	98
6.3.12.2	Visible attribute	98
6.3.12.3	Rect attribute	98
6.3.12.4	Bgcolor attribute	98
6.3.12.5	Bgradient attribute	98
6.3.12.6	Borderwidth attribute	99
6.3.12.7	Bordercolor attribute	99

6.3.13	Do elements	99
6.3.13.1	Wtvml:do element	99
6.3.13.1.1	Title attribute	101
6.3.13.1.2	Focus attribute	101
6.3.13.1.3	Localsrc attribute	101
6.3.13.1.4	Icon attribute	101
6.3.13.1.5	Bglocalsrc attribute	101
6.3.13.1.6	Bgoffset attribute	101
6.3.13.1.7	Pos Attribute	101
6.3.13.1.8	Example	102
6.3.13.2	Do element	102
6.3.13.2.1	Font attribute	102
6.3.13.2.2	Bgcolor attribute	103
6.3.13.2.3	Bglocalsrc attribute	103
6.3.13.2.4	Bgoffset attribute	103
6.3.13.2.5	Align element	103
6.3.13.2.6	Localsrc attribute	103
6.3.13.2.7	Example	103
6.3.14	Table element	104
6.3.15	Tr element	105
6.3.16	Td element	106
6.3.17	Menu element	107
6.3.17.1	Id attribute	108
6.3.17.2	Visible element	108
6.3.17.3	Rect attribute	108
6.3.17.4	Align element	108
6.3.17.5	Rowgap attribute	108
6.3.17.6	Font attribute	108
6.3.17.7	Aacolor attribute	108
6.3.17.8	Bgcolor attribute	109
6.3.17.9	Bgradius attribute	109
6.3.17.10	Bordercolor attribute	109
6.3.17.11	Borderwidth attribute	109
6.3.17.12	Hpad attribute	109
6.3.17.13	Vpad attribute	109
6.3.17.14	Mode Attribute	109
6.3.17.15	Example 1	109
6.3.17.16	Example 2	109
6.3.18	Popupdo element	110
6.3.19	Popupmenu element	111
6.3.19.1	Pos attribute	112
6.3.19.2	Pivot attribute	112
6.3.19.3	Size attribute	113
6.3.19.4	Minsize attribute	113
6.3.19.5	Maxsize attribute	113
6.3.19.6	Align attribute	113
6.3.19.7	Example	113
6.3.20	Scroll element	114
6.3.20.1	Fgcolor attribute	114
6.3.20.2	Fgfocused attribute	114
6.3.20.3	Bgcolor attribute	114
6.3.20.4	Bgfocused attribute	114
6.3.20.5	Example	114
6.3.21	Select element	115
6.3.21.1	Bgcolor attribute	116
6.3.21.2	Bordercolor attribute	116
6.3.21.3	Borderfocused attribute	116
6.3.21.4	Style attribute	116
6.3.21.5	Bgradius attribute	116
6.3.21.6	Borderwidth attribute	116
6.3.21.7	Pivot attribute	116
6.3.21.8	Example	116

6.3.22	Optgroup element	117
6.3.22.1	Bgcolor attribute	117
6.3.22.2	Bordercolor attribute	117
6.3.22.3	Borderfocused attribute	117
6.3.22.4	Bgradius attribute	118
6.3.22.5	Borderwidth attribute	118
6.3.22.6	Example	118
6.3.23	Input element	118
6.3.23.1	Cursorcolor attribute	119
6.3.23.2	Bgcolor attribute	119
6.3.23.3	Borderwidth attribute	119
6.3.23.4	Bordercolor attribute	119
6.3.23.5	Borderfocused attribute	119
6.3.23.6	Bgradius attribute	119
6.3.23.7	Borderwidth attribute	120
6.3.23.8	Example	120
6.3.24	Dialog element	120
6.3.24.1	Id attribute	121
6.3.24.2	Visible element	121
6.3.24.3	Rect attribute	122
6.3.24.4	Aacolor attribute	122
6.3.24.5	Bgcolor attribute	122
6.3.24.6	Bordercolor attribute	122
6.3.24.7	Borderfocused attribute	122
6.3.24.8	Bgradius attribute	122
6.3.24.9	Borderwidth attribute	122
6.3.24.10	Example	122
6.3.25	Browser element	123
6.3.25.1	Id attribute	124
6.3.25.2	Type attribute	124
6.3.25.3	Visible attribute	124
6.3.25.4	Rect attribute	125
6.3.25.5	Aacolor attribute	125
6.3.25.6	Bgcolor attribute	125
6.3.25.7	Bordercolor attribute	125
6.3.25.8	Borderfocused attribute	125
6.3.25.9	Bgradius attribute	125
6.3.25.10	Borderwidth attribute	125
6.3.25.11	Example	125
6.3.26	Mode element	127
6.3.26.1	Id attribute	128
6.3.26.2	Bgimage attribute	128
6.3.26.3	Translucency attribute	128
6.3.26.4	Example	129
6.3.27	Modeset element	130
6.3.27.1	Fontsrc attribute	130
6.3.27.2	Localsrc attribute	130
6.3.27.3	Dosrc attribute	131
6.3.27.4	Example	131
7	WTVML 6.3 Fontset specification	131
7.1	Cache control	131
7.1.1	Head element	132
7.1.2	Meta element	132
7.1.2.1	Http-equiv attribute	132
7.1.2.2	Content attribute	132
7.2	Fonts and fontsets	133
7.2.1	Font element	133
7.2.1.1	Id attribute	133
7.2.1.2	Src attribute	134
7.2.1.3	Size attribute	134
7.2.1.4	Fgstatic attribute	134

7.2.1.5	Fgcolor attribute	134
7.2.1.6	Fgfocused attribute.....	134
7.2.1.7	Fgactivated attribute.....	134
7.2.1.8	Example	134
7.2.2	Fontset element.....	135
7.2.2.1	Example	135
8	WTVML 6.3 Doset specification	136
8.1	Wtvml:do elements.....	136
8.2	Cache control.....	136
8.2.1	Head element	136
8.2.2	Meta element	137
8.2.2.1	Http-equiv attribute	137
8.2.2.2	Content attribute.....	137
8.2.3	Wtvml:do element	137
8.2.3.1	Type attribute	138
8.2.3.2	Label attribute	138
8.2.3.3	Name attribute.....	139
8.2.3.4	Tabindex attribute	139
8.2.3.5	Title attribute.....	139
8.2.3.6	Localsrc attribute.....	139
8.2.4	Doset element	139
8.2.4.1	Doset example file.....	140
9	Document Object Model	140
9.1	Setting object attribute values	141
9.1.1	Using the '&' character in setattr statements	141
9.2	Retrieving attribute values.....	141
9.3	Wtv object	141
9.3.1	Visible attribute	141
9.3.2	Version attribute	141
9.3.3	Mode attribute.....	141
9.3.4	Mode.fragment attribute	142
9.3.5	Locale attribute	142
9.3.6	Video attribute	142
9.3.7	Audio attribute	142
9.3.8	Alert attribute.....	142
9.3.9	Timer attribute	142
9.3.10	Flushcache attribute	142
9.3.11	Rand attribute.....	143
9.3.12	Launch attribute	143
9.3.13	Stb attribute.....	143
9.3.14	Xml-debug attribute.....	143
9.3.15	Mem-osd attribute.....	144
9.4	Http element.....	144
9.4.1	Connectstate attribute	144
9.4.2	Title attribute	144
9.4.3	Hint attribute	144
9.4.4	Connectdelay attribute	144
9.4.5	Tpsec attribute	145
9.4.6	Example 1	145
9.4.7	Example 2	145
9.5	URL syntax	145
9.5.1	Browser services.....	145
9.5.2	Blank: URL.....	146
10	WML script	146
10.1	Introduction	146
10.2	Base WML Script specification.....	146
10.3	WMLScript for the WTVML 6.3 microbrowser	146
10.3.1	GetAttr function.....	146
10.3.1.1	Example	146
10.3.2	Setattr function.....	147

10.3.2.1	Example	147
10.3.3	GoTarget function.....	147
10.3.3.1	Example	147
10.3.4	GetVar function	147
10.3.4.1	Example	147
10.3.5	SetVar function.....	147
10.3.5.1	Example	147
11	Validating markup.....	148
11.1	Validating wtvml files	148
11.1.1	Internal DTD subset.....	148
11.1.2	Root element.....	148
11.2	Validating modeset files	149
11.2.1	Root element.....	149
11.2.2	Do elements	149
11.3	Validating fontset files	150
11.4	Validating doset files.....	150
11.4.1	Root element.....	150
Annex A (informative):	Bibliography.....	152
History		153

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECTrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

Introduction

The WTVML Specification is a televisual, internet compatible content authoring format based on XML. It has all the required elements, attributes and events necessary to describe the user interface and user interaction models for the majority of Interactive TV services.

The format supports key features to enable 'service oriented' applications and also provides a good balance between the explicit layout and design requirements necessary for television and flexibility and dynamic capabilities of the Internet. Uniquely, the format fully describes the rendering and interaction of the Interactive TV service in such a manner as to allow its implementation in a number of different mechanisms and technologies, essentially delivering a high degree of portability to applications developed in the format. The format also assumes a user agent behaviour that is more sophisticated and more stateful than a standard internet based HTML browser, making it ideal for use in environments where a real time network connection is either not always available or is of a relatively low bandwidth.

The format combines the best of explicit "pixel perfect" control required for TV user interfaces, and the dynamic layout and internet compatibility requirements necessary for e-business and dynamic content. It provides a good separation between the layout of high level objects, and the dynamic layout of elements in the main mark-up language. This separation also allows for the default attribute values to be set for a given layout, effectively creating sets of inherited styles. One useful by product of this separation is the ability to deploy the service in a functionally identical way by interoperating differently or replacing the high level layout for different device types.

The format assumes a rich event model, and contains explicit state and variable management, allowing sophisticated user interface effects to be created without the use of scripting.

The present document contains the exhaustive set of elements and attributes that can be used to author a WTVML application. The present document also makes references in many places to the "browser's behaviour". It is common to implement WTVML functionality by way of a "microbrowser" and "gateway" architecture. In this architecture, the gateway processes the raw WTVML and generates compiled "bytecode" to be passed to the user agent to execute. It is expected that different networks may implement their own output formats from the gateway, consisting of WTVML bytecodes extended from the standard WAP Forum byte codes, HTML and Javascript™, Java execution classes or even code that can be run directly on legacy set-top boxes middleware or APIs.

In all instances where the term "the browser" is used in the present document, it is take to mean the combination of the gateway and client devices operating together to perform the function of a "browser" from the author's point of view.

In many places references are made in the present document to previous versions of WTVML or even to the original WML 1.3 or 1.2.1 specification. These are preserved in order for the industry to consider backwards compatibility issues with content already authored and available using early (non-public) versions of the WTVML specification.

In all cases where third-parties are implementing the present document, functionality documented as "Version 6" or above should be considered to be "standard".

WTVML is "backwards compatible" with WML 1.3, in such a way that given a valid WTVML mode context and style, and WML 1.3 services can be displayed in a WTVML browser object and menu object.

As such, textual content in the WTVML browser object is often in the present document referred to as WML as well as WTVML.

The reader of the present document should have all necessary information to develop WTVML compatible products and services. This would include:

- native browser implementations of the WTVML standard for various devices;
- gateway implementations of the WTVML standard, driving a variety of display mechanisms and formats;
- Tooling and service authoring frameworks, needing to output WTVML to publish their services to compatible browsers and networks;
- Individual services authors, wishing to develop in WTVML directly.

The present document is intended to be an exhaustive description of the format. Individuals wanting to learn the format should also consider reading training material and internet resources on the subject.

A list of WTVML related material, as well as an online developer's forum is available on line at www.waptv.com

1 Scope

The present document is the initial public release of WTVML. WTVML is an evolving standard designed to provide a device independent means to author service applications for televisions and other similarly capable small foot-print devices.

The growth of Digital TV receivers around the world is probably the most significant area of growth in digital devices, as PC based Internet and Mobile devices all reach maturity or saturation.

The need for a common authoring format for Interactive TV services is therefore clear, and yet the market is moving too quickly for it to stop and re-define a new format for television. It is with this in mind, and the belief that all Digital TV operators (including BSkyB) will benefit from a growth and in standardization of the Interactive Services marketplace on a pan-national scale, that this tried and tested format is submitted to the Industry.

WTVML is a mature markup language based on XML. It is a derivation of WML 1.3 as defined as part of the OMA's WAP 1.2.1 specifications, and is designed to deliver and specify content and user interfaces for Digital TV set-top-boxes and other similarly specified devices.

WTVML is designed with the constraints of the television in mind: These constraints include:

- Relatively small display and limited input facilities.
- Narrow band or absent network connection (return path).
- Limited memory and computational resources.
- The need for precise televisual control over positioning and style.
- The need to deliver consistent user interface behaviours and minimize the authoring effort to achieve them.

WTVML contains the following major functional areas:

- Dynamic text presentation and layout, including images support, formatting and sophisticated layout commands.
- Overall screen and style control, allowing the definition of screen templates (Modes) involving various high level objects types such as AV windows, browsers, menus, images etc.
- Sophisticated navigation model, where content can be organized into Cards and Decks. Cards specify one or more units of user interaction. Cards can be grouped into decks. A WTVML deck is similar to an HTML page in that it is defined by a URL (see RFC 2396 [2]) and is the unit of content transmission. Decks can reference different "Modes", as well as other resources.
- WTVML includes support for explicitly managing the navigation between cards and decks. WTVML also includes provisions for event handling in the device, which may be used for navigational purposes or to execute scripts or reset or change any attribute in the DOM or mode. WML also supports anchored links, similar to those found in HTML 4.0 Specification [1].
- String parameterization and state management - all WTVML decks can be parameterized using a state model. Variables can be used in the place of strings and are substituted at run-time. This parameterization allows for more efficient use of network resources, and very dynamic templated applications to be created.

WTVML can completely describe the user interface look, feel behaviour, and state models of an Interactive TV application. WTVML does this due to its sophisticated DTD and defined User Agent behaviour. Application UI effects are created by manipulating the WTVML elements and attributes directly, and rarely with the use of Script. This ensures that WTVML applications are "portable" with the fixed behaviour mapping well onto a number of set-top-box run time environments, through fixed transformations or compile steps.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

[1] HTML 4.0 Specification, W3C Recommendation (1997) REC-HTML40-971218.

NOTE: See <http://www.w3.org/TR/REC-html40>.

[2] IETF RFC 2396 (1998): "Uniform Resource Identifiers (URI): Generic Syntax".

[3] ETSI EN 300 468 (V1.5.1): "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

3 Definitions, abbreviations and typographical conventions

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

author: person or program that writes or generates WTVML, WMLScript or other content

client: device (or application) that initiates a request for a connection with a server

content: subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent in response to a user request

content encoding: when used as a verb, content encoding indicates the act of converting content from one format to another

NOTE: Typically the resulting format requires less physical space than the original, is easier to process or store and/or is encrypted. When used as a noun, content encoding specifies a particular format or encoding standard or process. WTVML is suitable for content encoding where the resultant content format has a direct relation to the original WTVML elements (for example tokenization), or where the resultant content format is a completely different language or output type (for example HTML/JavaScript™).

content format: actual representation of content

device: network entity that is capable of sending and receiving packets of information and has a unique device address

NOTE: A device can act as both a client or a server within a given context or across multiple contexts. For example, a device can service a number of clients (as a server) while being a client to another server.

JavaScript™: *de facto* standard language that can be used to add dynamic behaviour to HTML documents

NOTE: JavaScript™ is one of the originating technologies of ECMAScript.

man-machine interface: a synonym for user interface

origin server: server on which a given resource resides or is to be created

NOTE: Often referred to as a web server or an HTTP server.

resource: network data object or service that can be identified by a URL

NOTE: Resources may be available in multiple representations (e.g. multiple languages, data formats, size and resolutions) or vary in other ways.

server: device (or application) that passively waits for connection requests from one or more clients

NOTE: A server may accept or reject a connection request from a client.

terminal: device providing the user with user agent capabilities, including the ability to request and receive information

NOTE: Also called a Set-top-box or iDTV device.

user: user is a person who interacts with a user agent to view, hear, or otherwise use a resource

user agent: user agent is any software or device that interprets WTVML, WMLScript, or other resources

NOTE: This may include textual browsers, network edge transcoders, search engines, etc.

WMLScript: scripting language used to program the WTVML device

NOTE: WMLScript is an extended subset of the JavaScript™ scripting language.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
MIME	Multipurpose Internet Mail Extensions
OSD	On Screen Display
OLS	OnLine Server
PC	Personal Computer
STB	Set Top Box
UI	User Interface
URL	Uniform Resource Locator
USL	Universal Service Locator (a short cut to a URL under development)
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WML	Wireless Markup Language
WTML	Wireless Television Markup Language
XML	eXtensible Markup Language

3.3 Typographical conventions

In the present document, fixed-width courier font, is used to designate references to WML elements and attributes and XSD fragments:

```
<!--***wtvml additions made to go element: setattr subelement added***-->
<!--***wtvml exception made to go element: sendreferer, cache control attributes deleted*** -->
<!ELEMENT go (postfield | setvar | setattr)*>
<!ATTLIST go
  href "HREF"; #REQUIRED
  method (post | get) "get"
  enctype %ContentType; "application/x-www-form-urlencoded"
  accept-charset CDATA #IMPLIED
  target CDATA #IMPLIED
  %coreattrs;
```


In the present document, fixed-width courier font is used to designate code samples:

```
<tr>
  <td>
<input name="inpPostcode" title="Enter your full postcode with or without a space." maxlength="20"
size="5" format="*M*" type="text"/>
<img localsrc='blank' src='' alt='' width='5'  />
  </td>
  <td>
<input name="inpHouseNo" title="Enter numeric house number" maxlength="30" size="5" format="*N"
type="text"/>
  </td>
  <td>
<img localsrc='blank' src='' alt='' width='5'  />OR<img localsrc='blank' src='' alt='' width='5'
 /></td>
  <td>
<input name="inpHouseNo" title="Enter your house name" maxlength="30" size="14" format="*M"
type="text"/>
  </td>
</tr>
```

4 WTVML XSDs

4.1 Introduction

The WTVML makes use of four distinct XSDs that together define the appearance and behaviour of wtvml content and onscreen elements for every WTVML based service.

These are:

- Wtvml_[version_number].xsd;
- Modeset_[version_number].xsd;
- Fontset_[version_number].xsd;
- Doset_[version_number].xsd.

The relationship between these files is as follows:

A valid WTVML file must conform to wtvml_[version_number].xsd. The WTVML rules are a superset of the WAP Forum WML 1.3 Specification rules. The WTVML file points to a mode definition using the mode attribute of the wml element. The mode reference is a HREF pointing to a modeset XML file. If there is no mode defined, the Browser should use the mode from the previous card. If there is no previous card (i.e. the browser has just been instantiated) then the Browser should exit.

The modeset file is an XML file conforming to modeset[version_number].xsd. The modeset includes one or more mode(s) which define the screen layout and appearance characteristics for the service. A modeset file can in turn include references to a fontset and a doset. These references are defined in the fontsrc and dosrc attributes of the root modeset element. These attribute values are HREFs, pointing to external XML files structured according to fontset_[version_number].xsd and doset_[version_number].xsd respectively. The fontset file defines the characteristics of font resources to be used in this service. The doset file defines the top-level do objects to be included in the menu for this service.

The fontset_[version_number].xml and doset_[version_number].xml files are defined and maintained separately from the modeset.xml file and from each other, to enable both menu items and fonts to be defined at a portal level, and the settings to be easily inherited by individual services within a portal simply by pointing to a common fontset and doset from the service's modeset XML file.

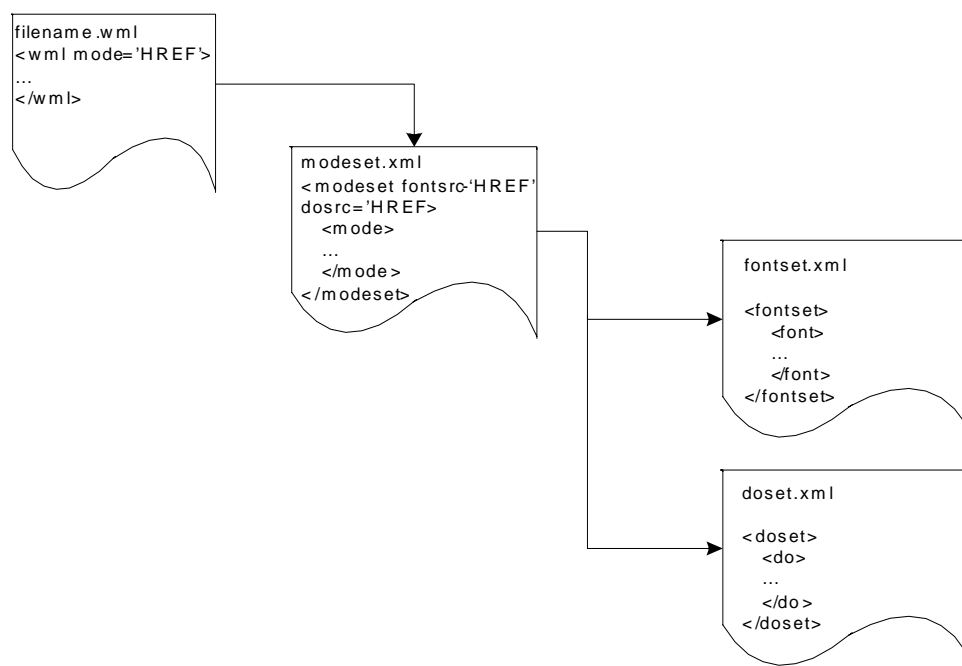


Figure 1: WTVML XSD file relationship

The wtvml XSD defines the characteristics of WML content for display using a WTVML 6.3 compliant microbrowser.

WTVML 6.3 is a superset of the WAP Forum WML 1.3 markup language. WML content originally designed for wireless delivery to mobile phone handsets will work without modification on waptv compliant microbrowsers, with the exception that any WBMP images will not appear on the Sky Interactive waptv microbrowser.

Many WML constructs have been extended or modified for use within the waptv environment. There are also some exceptions from the WML 1.3 specification.

All WTVML extensions, modifications and exceptions to the WML 1.3 standard are detailed in the present document. Unchanged elements are not detailed in the present document.

Developers should look to the WAP Forum WML Specification 1.3, available from the WAP Forum website at www.wapforum.com, for details of these unchanged elements.

5 WTVML specification

5.1 Decks and cards

5.1.1 WML element

diagram	<p>The diagram shows the structure of the WTVML element. The root element 'wml' contains a sequence of four child elements: 'wtvml:head', 'wtvml:template', 'wtvml:precache', and 'wtvml:card'. 'wtvml:head' is optional (0..1). 'wtvml:template' is optional (0..1) and is repeated in every card. 'wtvml:precache' is optional (0..∞) and controls cache behaviour. 'wtvml:card' is optional (1..∞) and is the WML card element.</p>																									
namespace	http://waptv.com/xsd/wtvml																									
children	wtvml:head wtvml:template wtvml:precache wtvml:card																									
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>mode</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>id</td> <td>xs:ID</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>class</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>update</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>auto</td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	mode	xs:string	optional			id	xs:ID	optional			class	xs:string	optional			update	xs:NMTOKEN	optional	auto	
Name	Type	Use	Default	Fixed																						
mode	xs:string	optional																								
id	xs:ID	optional																								
class	xs:string	optional																								
update	xs:NMTOKEN	optional	auto																							
source	<pre> <xs:element name="wml"> <xs:annotation> <xs:documentation>root element</xs:documentation> <xs:appinfo> <comp:element-def name="wml" bytecode="3f" isroot="true"/> <comp:allowed-attribute name="xmlns" discard="true"/> <comp:allowed-attribute name="xmlns:comp" discard="true"/> <comp:allowed-attribute name="xmlns:xsi" discard="true"/> <comp:allowed-attribute name="xsi:schemaLocation" discard="true"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="wtvml:head" minOccurs="0"/> <xs:element ref="wtvml:template" minOccurs="0"/> <xs:element ref="wtvml:precache" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:card" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute name="mode" type="xs:string" use="optional"/> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> <xs:attribute name="update" use="optional" default="auto"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="auto"/> <xs:enumeration value="oncardnav"/> <xs:enumeration value="noauto"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>																									

5.1.1.1 Mode attribute

mode="http://mysite/modeset.xml"

The mode attribute defines the name and location of the modeset XML file to be used by the browser when displaying the WML content of the current deck. The modeset can include multiple individual modes. A relative or absolute URL can be used for this attribute.

5.1.1.2 Update attribute

update=(auto | oncardnav | noauto)

The update attribute can be used to control the update behaviour for content in a dynamic broadcast content churn.

Value	Description
auto	Browser should perform an automatic screen refresh as soon as it detects new content in the broadcast stream. This means displayed can change without any user interaction. This is the default value for the update attribute.
oncardnav	Browser should refresh content when the browser is navigated to another card, including cards in the same deck.
noauto	Browser should only refresh when it is navigated to a new deck.

Default value for update is "auto".


5.1.1.3 Attributes defined elsewhere

The id, class and update attributes are standard WAP Forum WML and are defined in the WML Specification 1.3 available from the WAP Forum at <http://www.wapforum.com>

5.1.1.4 Example

```
<wml mode="http://mysite/modeset.xml">
<head>
  <meta http-equiv="Cache-Control" content="no-cache" />
</head>
<template>
  <do type="vnd.wtv-g" label="xxxx">
    <go href="http://xyz.org/xxzz" />
  </do>
</template>
</wml>
```

5.1.2 Precache element

diagram																					
namespace	http://waptv.com/xsd/wtvml																				
Used by	element wml																				
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>href</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>cache-set</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>cache-control</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>cache</td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	href	xs:string	required			cache-set	xs:string	optional			cache-control	xs:NMTOKEN	optional	cache	
Name	Type	Use	Default	Fixed																	
href	xs:string	required																			
cache-set	xs:string	optional																			
cache-control	xs:NMTOKEN	optional	cache																		
source	<pre><xs:element name="precache"> <xs:annotation> <xs:documentation>controls cache behaviour</xs:documentation> <xs:appinfo> <comp:element-def name="precache" bytecode="105"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="href" type="xs:string" use="required"/> <xs:attribute name="cache-set" type="xs:string" use="optional"/> <xs:attribute name="cache-control" use="optional" default="cache"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="cache"/> <xs:enumeration value="no-cache"/> <xs:enumeration value="domain"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element></pre>																				

The precache element can be used to precisely control browser caching behaviour. Using this element, it is possible to selectively cache and flush resources. Typically this is used to control the downloading, caching and flushing of graphic image resources to optimize user experience.

Browser behaviour is to first download the WTVML deck containing the root card, along with a modeset and any other assets required. Only then will any precache directives be actioned.

There is potential for subsequent card loads from new decks to be slower, as there is potential for the download of a subsequent deck to be sharing bandwidth with any precache downloads already occurring.

Application authors are advised to design "just in time" precache behaviour into their WTVML services to avoid potential slowing of subsequent deck download, and to avoid potential cache overflow.

5.1.2.1 Href attribute

```
href="http://mysite/myimage.jpg"
```

The href attribute specifies the URL of the resource to be precached.

Individual precache requests are added to a queue in the order of their declaration, and are fetched one at a time while the browser is idle.

5.1.2.2 Cache-set attribute

```
cache-set="a"
```

The cache-set attribute is used to assign a single character identifier to the cached resource. This can be any ASCII character except those used in the standard WML control character set.

The same character identifier can be assigned to multiple resources.

The default value for the cache-set attribute is 0.

It is possible to selectively flush cached resources according to their cache-set attributes by using <setattr wtv.flushcache>.

See clause 9.3.10 for details on using the wtv.flushcache attribute.

5.1.2.3 The cache-control attribute

cache-control="(domain|cache|nocache)"

The cache-control attribute specifies how the browser should cache this resource. There are three possible values for the cache-control attribute. The default value is "cache".

Value	Description
Domain	Keep the resource in cache until a domain change or until the memory is required by the browser.
Cache	Keep the resource in cache until the memory is required by the browser.
No-cache	Do not cache the resource.

5.1.3 Void

5.1.4 Card element

diagram																																																																																						
namespace	http://waptv.com/xsd/wtvml																																																																																					
children	wtvml:onevent wtvml:timer wtvml:do wtvml:p																																																																																					
used by	element wml																																																																																					
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>newcontext</td> <td>xs:boolean</td> <td>optional</td> <td>false</td> <td></td> </tr> <tr> <td>ordered</td> <td>xs:boolean</td> <td>optional</td> <td>true</td> <td></td> </tr> <tr> <td>mode</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>paragap</td> <td>xs:integer</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>scroll</td> <td>xs:boolean</td> <td>optional</td> <td>false</td> <td></td> </tr> <tr> <td>tabindex</td> <td>xs:integer</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>onenterforward</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>onenterbackward</td> <td>xs:string</td> <td>optional</td> <td>optional</td> <td></td> </tr> <tr> <td>ontimer</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>id</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>class</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>vspace</td> <td>xs:integer</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>hspace</td> <td>xs:integer</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>hint</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>navigation</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>default</td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	title	xs:string	optional			newcontext	xs:boolean	optional	false		ordered	xs:boolean	optional	true		mode	xs:string	optional			paragap	xs:integer	optional			scroll	xs:boolean	optional	false		tabindex	xs:integer	optional			onenterforward	xs:string	optional			onenterbackward	xs:string	optional	optional		ontimer	xs:string	optional			id	xs:string	optional			class	xs:string	optional			vspace	xs:integer	optional			hspace	xs:integer	optional			hint	xs:string	optional			navigation	xs:NMTOKEN	optional	default	
Name	Type	Use	Default	Fixed																																																																																		
title	xs:string	optional																																																																																				
newcontext	xs:boolean	optional	false																																																																																			
ordered	xs:boolean	optional	true																																																																																			
mode	xs:string	optional																																																																																				
paragap	xs:integer	optional																																																																																				
scroll	xs:boolean	optional	false																																																																																			
tabindex	xs:integer	optional																																																																																				
onenterforward	xs:string	optional																																																																																				
onenterbackward	xs:string	optional	optional																																																																																			
ontimer	xs:string	optional																																																																																				
id	xs:string	optional																																																																																				
class	xs:string	optional																																																																																				
vspace	xs:integer	optional																																																																																				
hspace	xs:integer	optional																																																																																				
hint	xs:string	optional																																																																																				
navigation	xs:NMTOKEN	optional	default																																																																																			
source	<pre> <xs:element name="card"> <xs:annotation> <xs:appinfo> <comp:element-def name="card" bytecode="27"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtvml:onevent" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:timer" minOccurs="0"/> <xs:element ref="wtvml:do" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:p" maxOccurs="unbounded"/> </xs:choice> <xs:attribute name="title" type="xs:string" use="optional"/> <xs:attribute name="newcontext" type="xs:boolean" use="optional" default="false"/> <xs:attribute name="ordered" type="xs:boolean" use="optional" default="true"/> <xs:attribute name="mode" type="xs:string" use="optional"/> <xs:attribute name="paragap" type="xs:integer" use="optional"/> <xs:attribute name="scroll" type="xs:boolean" use="optional" default="false"/> <xs:attribute name="tabindex" type="xs:integer" use="optional"/> <xs:attribute name="onenterforward" type="xs:string" use="optional"/> <xs:attribute name="onenterbackward" type="xs:string" use="optional"/> <xs:attribute name="ontimer" type="xs:string" use="optional"/> <xs:attribute name="id" type="xs:string" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> <xs:attribute name="vspace" type="xs:integer" use="optional"/> <xs:attribute name="hspace" type="xs:integer" use="optional"/> <xs:attribute name="hint" type="xs:string" use="optional"/> <xs:attribute name="navigation" use="optional" default="default"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tabindex"/> <xs:enumeration value="default"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>																																																																																					

5.1.4.1 Mode attribute

`mode="mymode"`

The mode attribute defines the mode within the current modeset to be used when displaying the content of the current card.

The value of the mode attribute corresponds to the id of the individual mode within the active modeset that is to be used when rendering this card for viewing.

See clause 6.3.26 for details on defining modes.

5.1.4.2 Paragap attribute

`paragap=n`

The paragap attribute defines the number of pixels that should be placed between paragraph elements within the card.

The default spacing is relatively close. On a detailed full screen of text, legibility can be increased by defining a paragap value.

5.1.4.3 Scroll attribute

`scroll="true|false"`

The scroll attribute defines whether or not the browser should pre-allocate space for a scroll bar as it lays out a WML card on screen.

Default behaviour is to attempt to lay out the card without a scroll bar. If the content proves to be too large for the screen, the Browser should add the scrollbar and lay out the content again.

The application author can reduce the time it takes to lay out a lengthy WML card by using the scroll attribute.

5.1.4.4 Tabindex attribute

`tabindex=-1`

The tabindex attribute is not a part of the card element in standard WML. In WTVML 6.3, a tabindex attribute of -1 on a card element can be used to send focus from the WML card to the Menu.

5.1.4.5 Hint attribute

`hint="vdata"`

The hint attribute is used to specify a text string to appear in the hint text object when the card is loaded. For most WML content, the title attribute is used to deliver this functionality. The card element is an exception because its title attribute is used to populate the title object.

5.1.4.6 Hspace attribute

`hspace="10"`

The hspace attribute specify in pixels the gap to be left between the left and right edges of the browser and the WML content.

5.1.4.7 Vspace attribute

`vspace="10"`

The vspace attribute specify in pixels the gap to be left between the top and bottom edges of the browser and the WML content displayed.

5.1.4.8 Navigation attribute

navigation="default|tabindex"

The navigation attribute can be used to achieve fine control over the focus navigation behaviour in complex card layouts.

Where the navigation attribute has been set to the value "tabindex", arrow button focus navigation will be controlled as follows:

Arrow button	Focus behaviour
Up	Focus moves to element with next lowest tabindex value.
Left	Focus moves to element with next lowest tabindex value.
Down	Focus moves to element with next highest tabindex value.
Right	Focus moves to element with next highest tabindex value.

5.1.4.9 Example

```
<wml mode="layout.xml" >
<head>
  <meta http-equiv="Cache-Control" content="no-cache" />
</head>
<template>
  <do type="vnd.wtv-g" label="xxxx">
    <go href="http://xyz.org/xxzz" />
  </do>
</template>
<card id="card1" title="Welcome" newcontext="true" mode="#MODE_ID" paragap="10" >
<p>
  Hello world!
</p>
<p>
  First sample WML
</p>
</card>
</wml>
```

5.2 Event bindings

5.2.1 Onevent element

diagram						
namespace	http://waptv.com/xsd/wtvml					
children	wtvml:go wtvml:prev wtvml:noop wtvml:refresh					
used by	elements card input option td template tr					
attributes	Name	Type	Use	Default	Fixed	Annotation
	type	xs:NMTOKEN		required		
	id	xs:ID	optional			
	class	xs:string	optional			
annotation	appInfo <comp:element-def name="onevent" bytecode="33"/>					
source	<pre> <xs:element name="onevent"> <xs:annotation> <xs:appinfo> <comp:element-def name="onevent" bytecode="33"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice> <xs:element ref="wtvml:go"/> <xs:element ref="wtvml:prev"/> <xs:element ref="wtvml:noop"/> <xs:element ref="wtvml:refresh"/> </xs:choice> <xs:attribute name="type" use="required"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="onfocus"/> <xs:enumeration value="onunfocus"/> <xs:enumeration value="onenter"/> <xs:enumeration value="onenterforwards"/> <xs:enumeration value="onenterbackwards"/> <xs:enumeration value="onactivate"/> <xs:enumeration value="ontimer"/> <xs:enumeration value="onpick"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>					


The onevent element is used to define an event to which an action can be bound.

5.2.1.1 Type attribute

There are eight onevent types supported within wtvml.

Type	Description
onfocus	Triggered when user navigates focus to this anchor.
onunfocus	Triggered when user navigates focus away from this anchor.
onenterforward	Triggered when user enters this card forwards.
onenterbackward	Triggered when user enters a card by using a URL retrieved from the browser history stack. Typically this will be via a prev task triggered by pressing the backup button.
onenter	A combination of onenterforward and onenterbackward. Triggered whenever the user enters this card, no matter how the URL was opened. Onenter is overridden where onenterforward or onenterbackward has already been defined.
onactivate	Triggered when user presses the select button to activate this anchor.
ontimer	Triggered when the timer expires.
onpick	Triggered when the user selects or deselects this option in a select list.

5.2.2 Timer element

diagram					
namespace	http://waptv.com/xsd/wtvml				
used by	element card				
attributes	Name	Type	Use	Default	Fixed
	name	xs:NMTOKEN	optional		
	value	xs:string	required		
	id	xs:ID	optional		
	class	xs:string	optional		
	retro	xs:NMTOKEN	optional	false	
source	<pre> <xs:element name="timer"> <xs:annotation> <xs:appinfo> <comp:element-def name="timer" bytecode="3c"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="name" type="xs:NMTOKEN" use="optional"/> <xs:attribute name="value" type="xs:string" use="required"/> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> <xs:attribute name="retro" use="optional" default="false"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="true"/> <xs:enumeration value="false"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>				

The timer element is used to trigger an ontimer event.

5.2.2.1 Value attribute

Version 6.0 of the WTVML standard enhances the standard WML timer element in two ways:

- Adds support for absolute timestamp values as well as tick values.
- Adds support for "time left" recording.

An absolute timestamp value can be specified using the form [YY:MM:DD:hh:mm:ss]. The timestamp must be 17 characters in length, otherwise it will be ignored. Zero padding can be used where required.

The absolute timestamp event will fire when the specified timestamp (maximum of one per card) expires with reference to the current set-top-box system time.

The Year, Month and Date can be defined as zero values, in which case the current system date will be used. The Hours, Minutes, Seconds values are compulsory.

The value for a card's timer can be set to a "tick" value as previously, or to an absolute timestamp value using the 17 character [YY:MM:DD:hh:mm:ss] notation. These formats are mutually exclusive.

The timer element records time left information to the "name" variable when a navigation event interrupts a running timer.

- `<timer name="mytimer" value="40"/>`

timer expires at 40 "ticks" (tenths of a second) after card load.

If a navigation event interrupts this timer before it fires, the remaining tick value can be retrieved using `$(mytimer)` or `$([browser_id].mytimer)`.

- `<timer value="02:12:14:00:08:05" />`

timer expires 14th December, 2002 at eight minutes past midnight and five seconds.

- `<timer value="00:00:00:07:42:34" />`

timer expires today, at forty-two minutes past seven am and thirty-four seconds.

5.2.2.2 Retro attribute

`retro=(true|false)`

The retro attribute is optional, and applies to timestamp values only. Default value for retro is "false".

When set to true, a timer with a timestamp value which has already expired when the containing card is loaded will still fire an ontimer event on card load.

The default behaviour is for the timer not to fire for a newly loaded expired timer.

- `<timer value="02:06:10:14:21:12" retro="true" />`

5.2.3 Do element

diagram																																																																																	
namespace	http://waptv.com/xsd/wtvml																																																																																
children	wtvml:go wtvml:prev wtvml:noop wtvml:refresh																																																																																
used by	elements b big card em fieldset font i p pre small strong td template u																																																																																
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>type</td> <td>xs:NMTOKEN</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>label</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>name</td> <td>xs:NMTOKEN</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>optional</td> <td>xs:boolean</td> <td>optional</td> <td>false</td> <td></td> </tr> <tr> <td>tabindex</td> <td>xs:NMTOKEN</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>title</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>focus</td> <td>xs:boolean</td> <td>optional</td> <td>true</td> <td></td> </tr> <tr> <td>id</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>class</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>icon</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>localsrc</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>font</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bglocalsrc</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bgoffset</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>pos</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	type	xs:NMTOKEN	required			label	xs:string	optional			name	xs:NMTOKEN	optional			optional	xs:boolean	optional	false		tabindex	xs:NMTOKEN	optional			title	xs:string	optional			focus	xs:boolean	optional	true		id	xs:string	optional			class	xs:string	optional			icon	xs:string	optional			localsrc	xs:string	optional			font	xs:string	optional			bglocalsrc	xs:string	optional			bgoffset	xs:string	optional			pos	xs:string	optional		
Name	Type	Use	Default	Fixed																																																																													
type	xs:NMTOKEN	required																																																																															
label	xs:string	optional																																																																															
name	xs:NMTOKEN	optional																																																																															
optional	xs:boolean	optional	false																																																																														
tabindex	xs:NMTOKEN	optional																																																																															
title	xs:string	optional																																																																															
focus	xs:boolean	optional	true																																																																														
id	xs:string	optional																																																																															
class	xs:string	optional																																																																															
icon	xs:string	optional																																																																															
localsrc	xs:string	optional																																																																															
font	xs:string	optional																																																																															
bglocalsrc	xs:string	optional																																																																															
bgoffset	xs:string	optional																																																																															
pos	xs:string	optional																																																																															
source	<pre> <xs:element name="do"> <xs:annotation> <xs:appinfo> <comp:element-def name="do" bytecode="28" /> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0"> <xs:element ref="wtvml:go" minOccurs="0" /> <xs:element ref="wtvml:prev" minOccurs="0" /> <xs:element ref="wtvml:noop" minOccurs="0" /> <xs:element ref="wtvml:refresh" minOccurs="0" /> </xs:choice> <xs:attribute name="type" use="required"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="accept" /> <xs:enumeration value="prev" /> <xs:enumeration value="help" /> <xs:enumeration value="reset" /> <xs:enumeration value="options" /> <xs:enumeration value="delete" /> <xs:enumeration value="unknown" /> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="label" type="xs:string" use="optional" /> <xs:attribute name="name" type="xs:NMTOKEN" use="optional" /> <xs:attribute name="optional" type="xs:boolean" use="optional" default="false" /> <xs:attribute name="tabindex" type="xs:NMTOKEN" use="optional" /> <xs:attribute name="title" type="xs:string" use="optional" /> <xs:attribute name="focus" type="xs:boolean" use="optional" default="true" /> <xs:attribute name="id" type="xs:string" use="optional" /> <xs:attribute name="class" type="xs:string" use="optional" /> <xs:attribute name="icon" type="xs:string" use="optional" /> <xs:attribute name="localsrc" type="xs:string" use="optional" /> <xs:attribute name="font" type="xs:string" use="optional" /> <xs:attribute name="bglocalsrc" type="xs:string" use="optional" /> <xs:attribute name="bgoffset" type="xs:string" use="optional" /> <xs:attribute name="pos" type="xs:string" use="optional" /> </xs:complexType> </xs:element> </pre>																																																																																

The do element provides the generic mechanism to allow a user to perform an action on a card.

5.2.3.1 Type attribute

type="(accept | prev | help | reset | options | delete | unknown | vnd.*)"

The generic WAP Forum WML doctype attribute is designed to provide a mechanism where the application author's intent can be conveyed to a user agent, which can then map the do to an appropriate interface widget.

In wtvml, the doctype attribute is used to associate an accelerator key with a do action.

Some dotypes are not associated with any specific button ("unknown", "reset", "options"). Do's with these dotypes can therefore only be actioned by navigating focus to them and pressing the Select button.

Dos that are not intended to be button-accelerated should always be defined with a doctype of "unknown".

Dotype	Description
Accept	The generic WML positive acknowledgement do type. Bound to the green button.
prev	Backward history navigation. Bound to the backup button. Browser keeps previous 20 navigations in history stack. On the 21 st backup button press, the Browser should load the "top:" URL. Top URL can be defined for each service in the Service Table.
help	Request for help. Bound to the help button.
reset	Generic WML do type for clearing or resetting a state. Not bound to any button.
options	Generic WML do type for options menu. Not bound to any button.
delete	The generic WML delete do type. Bound to the Yellow button.
unknown	Generic WML do type for other kinds of actions. Not bound to any button.
vnd.*	Vendor-specific or user-agent specific types. Users can map these dotypes onto available UI control widgets as required. Vendors should allocate names with the format vnd.[platform_name]-[doctype]. For example, within a specific platform, the doctype that specifies the "red button" on the remote control handset is designated as vnd.wtv-r.

5.2.3.2 Label attribute

The label attribute specifies the text label that should be associated with a button accelerated do when rendered on screen.

To render a button without a label, application authors should define a label value that consists of a single space.

```
<do ... label=" ">
```

5.2.3.3 Name attribute

Menu dos are sorted alphabetically by their name attribute. This allows application authors to customize the order of menu dos independently of the table and tabindex attributes.

5.2.3.4 Title attribute

```
title ="CDATA"
```

The title attribute is used to specify text content to be displayed in the hint text pane when this do has focus.

This provides a useful mechanism to deliver context-sensitive instructions and prompts to the user. For example, a do could be labelled "Send Message" and have a title of "Press Select to send your message now". The title text will only be visible when this do has focus, thus providing additional help to the user.

5.2.3.5 Focus attribute

focus="(true|false)"

The focus attribute is used to control whether or not the user can navigate focus to this do element. This can be useful to permit certain do actions to be activated only by explicit key binding actions.

In the following example, the Help card can only be accessed through using the dedicated "help" key on the remote control.

5.2.3.6 Void

5.2.3.7 Localsrc attribute

localsrc=CDATA

The localsrc attribute has been deprecated from the v6 microbrowser. From version 6.0 onward, the functionality previously provided by the localsrc attribute will now be provided by the icon attribute.

NOTE: The version 6 microBrowser should continue to honour the localsrc attribute to ensure backward compatibility. Developers of new services must use the new icon attribute in place of localsrc.

5.2.3.8 Icon attribute

The microBrowser should attach a localsrc image to most actionable dos. The image corresponds to the "doctype" specified for the do. For example, a do with a "doctype" of vnd.wtv-g will have the "green" button icon drawn onscreen beside the do.

By default, dos of "unknown" doctype will have the "a" icon image drawn on screen beside the do.

Application authors can replace the default image with another localsrc image, including images of their own design, or with a blank image, using the icon attribute.

To remove the default icon from the do, specify icon="blank", or icon="" in the do element.

5.2.3.9 Font attribute

font=CDATA

The font attribute can be used to specify a font for this do.

5.2.3.10 Bglocalsrc attribute

bglocalsrc=href

The bglocalsrc attribute is used to set a background image behind a menu do item.

This default setting can be overridden at card level if required, by defining a bglocalsrc attribute for a menu do.

5.2.3.11 Bgoffset attribute

bgoffset="x,y"

The bgoffset attribute defines any offset of a local source image behind a menu do element.

This default setting can be overridden at card level if required by defining a bgoffset attribute for a menu do.

5.2.3.12 Pos attribute

pos=%pos

The `pos` attribute defines the x,y location for a "floating" do.

Any do defined in wtvml at card or deck level that is not an inline do (i.e. that is not nested within `<p>` `</p>` tags) will appear in the menu, unless the `pos` attribute has been defined, in which case it will be a floating do and will be anchored onscreen at the x,y coordinates defined by the `pos` attribute.

5.2.3.13 Example

```
<wml mode="layout.xml">

<template>
  <do type="help" label="Help" focus="false">
    <go href="http://xyz.org/xx" />
  </do>

  <do type="vnd.wtv-g" label="xxxx" title="Press select for Information">
    <go href="http://xyz.org/xxzz" />
  </do>
</template>

<card id="card1" title="abc">
<p>
  xyz
</p>
</card>

</wml>
```


5.3 Tasks

5.3.1 Go element

diagram																																																			
namespace	http://waptv.com/xsd/wtvml																																																		
children	wtvml:postfield wtvml:setvar wtvml:setattr																																																		
Used by	elements anchor do onevent																																																		
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>href</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>method</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>get</td> <td></td> </tr> <tr> <td>enctype</td> <td>xs:string</td> <td>optional</td> <td>application/x-www-form-urlencoded</td> <td></td> </tr> <tr> <td>accept-charset</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>target</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>id</td> <td>xs:ID</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>class</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>cache-control</td> <td>xs:NMTOKEN</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>timeout</td> <td>xs:nonNegativeInteger</td> <td></td> <td>optional</td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	href	xs:string	required			method	xs:NMTOKEN	optional	get		enctype	xs:string	optional	application/x-www-form-urlencoded		accept-charset	xs:string	optional			target	xs:string	optional			id	xs:ID	optional			class	xs:string	optional			cache-control	xs:NMTOKEN	optional			timeout	xs:nonNegativeInteger		optional	
Name	Type	Use	Default	Fixed																																															
href	xs:string	required																																																	
method	xs:NMTOKEN	optional	get																																																
enctype	xs:string	optional	application/x-www-form-urlencoded																																																
accept-charset	xs:string	optional																																																	
target	xs:string	optional																																																	
id	xs:ID	optional																																																	
class	xs:string	optional																																																	
cache-control	xs:NMTOKEN	optional																																																	
timeout	xs:nonNegativeInteger		optional																																																
source	<pre> <xs:element name="go"> <xs:annotation> <xs:appinfo> <comp:element-def name="go" bytecode="2b" /> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtvml:postfield" minOccurs="0" maxOccurs="unbounded" /> <xs:element ref="wtvml:setvar" minOccurs="0" maxOccurs="unbounded" /> <xs:element ref="wtvml:setattr" minOccurs="0" maxOccurs="unbounded" /> </xs:choice> <xs:attribute name="href" type="xs:string" use="required" /> <xs:attribute name="method" use="optional" default="get"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="post" /> <xs:enumeration value="get" /> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="enctype" type="xs:string" use="optional" default="application/x-www-form-urlencoded" /> <xs:attribute name="accept-charset" type="xs:string" use="optional" /> <xs:attribute name="target" type="xs:string" use="optional" /> <xs:attribute name="id" type="xs:ID" use="optional" /> <xs:attribute name="class" type="xs:string" use="optional" /> <xs:attribute name="cache-control" use="optional"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="no-cache" /> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="timeout" type="xs:nonNegativeInteger" use="optional" /> </xs:complexType> </xs:element> </pre>																																																		

The go element has been modified to include a new sub element, setattr, and a new attribute, target. The WML 1.3 standard go attributes sendreferer and cache-control are exceptions from the WTVML 6.3 specification.

5.3.1.1 Setattr subelement

See clause 5.4.1 for information on the Setattr element.

5.3.1.2 Sendreferer attribute

sendreferer=boolean

The sendreferer attribute in the WML 1.3 specification is used to support a limited form of access control on WML content. In standard WML, if this attribute is true, the browser must specify the URI of the referring deck to the server. The server can then allow or deny the action depending on which deck made the request.

5.3.1.3 Target attribute

target="dialog"

The target attribute is used to specify the id of the browser object that the go should link to. For example, to display WML content in a popup dialog, the application author would assign a target of dialog to a go statement:

```
go href="popup.wml" target="dialog"
```

To load WML content into a subsidiary browser with an id of second the application author would assign a target of second to a go statement:

```
go href="myfile.wml" target="second"
```

If target is unspecified, the WML will load into the currently active browser.

If target specifies another browser, the focus will follow the WML into that other browser. Focus can be navigated between browsers using the arrow keys.

5.3.1.4 Cache-control attribute

cache-control=no-cache

If the cache-control value is present, and the value is "no-cache", the Browser should reload the URL from the origin server. This attribute represents the HTTP "cache-control" header.

5.3.1.5 Timeout attribute

timeout="10"

The timeout attribute is handled by the OLS, and allows the application author to override the 20 second default the OLS normally uses when requesting content from a webserver.

The timeout attribute value is measured in seconds.

5.3.1.6 Example

```
<wml>

<card id="card1" title="Card1">
  <do type="vnd.wtv-g" label="Popup" title="Press Green for Popup Browser">
    <go href="#popup" target="dialog">
      <setvar name="info" value="Coming from Card1" />
    </go>
  </do>

  <do type="vnd.wtv-b" label="Card2">
    <go href="#card2" target="dialog">
      <setvar name="info" value="You have come from Card1" />
    </go>
  </do>

<p>
  <big>Card1</big>
```

```

    <br/>$(info)
  </p>
</card>

<card id="card2" title="Card2" paragap="10">
  <p>
    $(info)
  </p>
  <p linegap="10">
    <anchor>Card1
      <go href="#card1">
        <setvar name="info" label="You have come from Card2" />
      </go>
    </anchor>
    <br/>
    <anchor>Popup
      <go href="#popup" target="dialog">
        <setvar name="info" label="You have come from Card2" />
      </go>
    </anchor>
  </p>
</card>

<card id="popup" title="Popup Browser" paragap="10">
  <do type="prev">
    <noop>
      <setvar name="info" value="You have come from Popup" />
    </noop>
  </do>

  <p>
    $(info)
    <br/>
    Press BackUp to go back OR Press Green for Card2.
  </p>

  <p>
    <do type="vnd.wtv-g" label="Card2" >
      <go href="#card2">
        <setvar name="info" value="You have come from Popup" />
      </go>
    </do>
  </p>
</card>

</wml>

```

5.3.2 Prev element

Diagram						
Namespace	http://waptv.com/xsd/wtvml					
Children	wtvml:setvar wtvml:setattr					
used by	elements anchor do onevent					
Attributes	Name	Type	Use	Default	Fixed	
	id	xs:ID	optional			
	class	xs:string	optional			
Source	<pre> <xs:element name="prev"> <xs:annotation> <xs:appinfo> <comp:element-def name="prev" bytecode="32"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtvml:setvar" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:setattr" minOccurs="0" maxOccurs="unbounded"/> </xs:choice> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>					

The prev element has been modified to include a new sub element, setattr.

5.3.2.1 Setattr subelement

See clause 5.4.2 for information on the Setattr element.

5.3.2.2 Example

```

<?xml version="1.0"?>

<wml mode='layout.xml'>

<head>
  <meta name="WTV_Connect_Now" content="" />
</head>

<template>
  <do type="prev">
    <setattr name='sample' value='' />
    <go href='home.wml' />
  </do>

  <do type='help' label=''>
    <go href='indexhelp.wml' />
  </do>
</template>

<card id="login" title='Welcome' paragap='17' focus="true">

<p>

  <do type='vnd.wtv-r' label='T&amp;Cs' title="Press Select to read Terms &amp; Conditions" focus>
    <go href='terms.wml' />
  </do>

</p>

<p>

```

Sample Code
 </p>
 </card>
 </wml>

5.3.3 Refresh element

diagram						
namespace	http://waptv.com/xsd/wtvml					
children	wtvml:setvar wtvml:setattr					
used by	elements anchor do onevent					
attributes	Name	Type	Use	Default	Fixed	
	id	xs:ID	optional			
	class	xs:string	optional			
source	<pre> <xs:element name="refresh"> <xs:annotation> <xs:appinfo> <comp:element-def name="refresh" bytecode="36"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtvml:setvar" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:setattr" minOccurs="0" maxOccurs="unbounded"/> </xs:choice> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>					

The refresh element has been modified to include a new sub element, setattr.

5.3.3.1 Settattr subelement

See clause 5.4.2 for information on the Setattr element.

5.3.4 Noop element

diagram						
namespace	http://waptv.com/xsd/wtvml					
children	wtvml:setvar wtvml:setattr					
used by	elements do onevent					
attributes	Name	Type	Use	Default	Fixed	
	id	xs:ID	optional			
	class	xs:string	optional			
source	<pre> <xs:element name="noop"> <xs:annotation> <xs:appinfo> <comp:element-def name="noop" bytecode="31"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtvml:setvar" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:setattr" minOccurs="0" maxOccurs="unbounded"/> </xs:choice> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>					

The noop element has been modified to include a new sub element, setattr.

5.3.4.1 Setattr subelement

See clause 5.4.2 for information on the Setattr element.

5.3.4.2 Example

```

<wml>

<card id="card1" title="Card1">
  <do type="vnd.wtv-g" label="Popup" title="Press Green for Popup Browser">
    <go href="#popup" target="dialog">
      <setvar name="info" value="Coming from Card1" />
    </go>
  </do>

  <do type="vnd.wtv-b" label="Card2">
    <go href="#card2" target="dialog">
      <setvar name="info" value="You have come from Card1" />
    </go>
  </do>

  <p>
    <big>Card1</big>
    <br/>$(info)
  </p>
</card>

<card id="card2" title="Card2" paragap="10">
  <p>
    $(info)
  </p>
  <p linegap="10">
    <anchor>Card1
      <go href="#card1">
        <setvar name="info" label="You have come from Card2" />
      </go>
    </anchor>

```

```

<br/>
<anchor>Popup
  <go href="#popup" target="dialog">
    <setvar name="info" label="You have come from Card2" />
  </go>
</anchor>
</p>
</card>

<card id="popup" title="Popup Browser" paragap="10">
  <do type="prev">
    <noop>
      <setvar name="info" value="You have come from Popup" />
    </noop>
  </do>

  <p>
    ${info}
    <br/>
    Press BackUp to go back OR Press Green for Card2.
  </p>


  <p>
    <do type="vnd.wtv-g" label="Card2" >
      <go href="#card2">
        <setvar name="info" value="You have come from Popup" />
      </go>
    </do>
  </p>
</card>

</wml>

```

5.4 Variables

5.4.1 Setvar element

diagram					
namespace	http://waptv.com/xsd/wtvm1				
used by	elements go noop prev refresh				
attributes	Name	Type	Use	Default	Fixed
	name	xs:string	required		
	value	xs:string	required		
	cache-control	xs:NMTOKEN	optional	context	
	id	xs:ID	optional		
	class	xs:string	optional		
source	<pre> <xs:element name="setvar"> <xs:annotation> <xs:appinfo> <comp:element-def name="setvar" bytecode="3e"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="name" type="xs:string" use="required"/> <xs:attribute name="value" type="xs:string" use="required"/> <xs:attribute name="cache-control" use="optional" default="context"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="domain"/> <xs:enumeration value="context"/> <xs:enumeration value="perm"/> <xs:enumeration value="delete"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

5.4.1.1 Name attribute

name="browsername.variablename"

Variables can be set across multiple browsers, and any browser's variables can be manipulated from any other browser.

EXAMPLE: Setvar name="second.myvar" value="World".

This will set the variable "myvar" to a value of "World" in the context of the browser with id "second".

If the browsername value is omitted from a setvar statement, then the variable is set in the context of the current browser.


5.4.1.2 Cache-control attribute

cache-control=context

Variables can be assigned a variety of lifetime characteristics:

- Domain: variables are only cleared on a domain change, and will survive a newcontext.
- Context: variables are only cleared on a newcontext.
- Perm: variables are only cleared by an explicit delete attribute and will survive a domain change and a newcontext.
- Domain & Context: variables are cleared by either a domain change or a newcontext.
- Delete: deletes the variable regardless of its lifetime flags.

5.4.2 Setattr element

diagram					
namespace	http://waptv.com/xsd/wtvmI				
used by	elements go noop prev refresh				
attributes	Name	Type	Use	Default	Fixed
	name	xs:string	required		
	value	xs:string	required		
	id	xs:ID	optional		
	class	xs:string	optional		
source	<pre><xs:element name="setattr"> <xs:annotation> <xs:appinfo> <comp:element-def name="setattr" bytecode="0d"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="name" type="xs:string" use="required"/> <xs:attribute name="value" type="xs:string" use="required"/> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> </xs:complexType> </xs:element></pre>				

The setattr element is used to set attributes for the various WTVML 6.3 onscreen objects.

Onscreen objects that are available for manipulation in this way include:

- Browser(s);
- The Audio Video object;
- The Title object;

- The callrate object;
- The status object;
- The USL object;
- The menu object;
- The dialog object;
- The browser object;
- The Image object;
- Localsrc objects.

See clause 9, The Document Object Model, for information on using the setattr statement.

5.4.2.1 Name attribute

```
name="wtv.mode"
```

The name attribute defines the onscreen object, and the object attribute to be modified using the syntax "objectid.attrname"

Most objects, including AV, dialog, title etc have static ids. The main browser object will always have the id "main" but subsidiary browsers can have any id.

When using setattr to load content into a browser, the name attribute defines the id of the target browser.

5.4.2.2 Value attribute

```
value="http://mydomain/mymode.xml"
```

The value attribute defines the value of the object attribute.

5.4.2.3 Examples

Using setattr to change mode:

- `<setattr name="wtv.mode" value="http://mydomain/mymode.xml"/>`
- `<setattr name="wtv.mode.fragment" value="#mode1"/>`

Using setattr to load a card into a browser:

- `<setattr name="main.href" value="http://mydomain/mydeck.wml#mycard"/>`
- `<setattr name="[browser_id].href" value="#mycard"/>`

5.4.3 Variable retrieval and setattr syntax

Variable values can be retrieved using dollar syntax as follows:

- `$([browser_id].[varname])`
- `$([varname])`
- `$([varname])`

Where a browser id is not specified, default behaviour is to retrieve the variable from the context of the local browser (i.e. the browser containing the card with the dollar syntax variable retrieval directive).

Regular brackets are conventional, but are not compulsory.

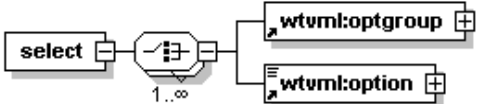
Attribute values can also be retrieved as follows:

- `$([object]![attribute_name])`

Code format	Description
<code>\$(var1)</code>	Var1 state variable in default browser.
<code>\$(main.href)</code>	Href attribute of main browser object.
<code>\$(main.var1)</code>	Var1 state variable in "main" browser.
<code>\$(second.var1)</code>	Var1 state variable in "second" browser.
<code>\$(main!href)</code>	refers to the href attribute of main browser.
<code>\$(wtv.mode.fragment)</code>	Fragment attribute of mode subobject of wtv object.
<code>\$(wtv.mode!fragment)</code>	refers to fragment attribute of mode subobject of wtv object.

5.5 Card fields

5.5.1 Select element

diagram																																																																												
namespace	http://waptv.com/xsd/wtvm1																																																																											
children	wtvml:optgroup wtvml:option																																																																											
used by	elements b big em fieldset font i p pre small strong td u																																																																											
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>name</td> <td>xs:NMTOKEN</td> <td></td> <td></td> <td></td> </tr> <tr> <td>value</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>iname</td> <td>xs:NMTOKEN</td> <td></td> <td></td> <td></td> </tr> <tr> <td>ivalue</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>size</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>width</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>multiple</td> <td>xs:boolean</td> <td></td> <td>false</td> <td></td> </tr> <tr> <td>tabindex</td> <td>xs:NMTOKEN</td> <td></td> <td></td> <td></td> </tr> <tr> <td>id</td> <td>xs:ID</td> <td></td> <td></td> <td></td> </tr> <tr> <td>class</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>localsrc</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>style</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>normal</td> <td></td> </tr> <tr> <td>pivot</td> <td>xs:NMTOKEN</td> <td>optional</td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	title	xs:string				name	xs:NMTOKEN				value	xs:string				iname	xs:NMTOKEN				ivalue	xs:string				size	xs:string				width	xs:string				multiple	xs:boolean		false		tabindex	xs:NMTOKEN				id	xs:ID				class	xs:string				localsrc	xs:string	optional			style	xs:NMTOKEN	optional	normal		pivot	xs:NMTOKEN	optional		
Name	Type	Use	Default	Fixed																																																																								
title	xs:string																																																																											
name	xs:NMTOKEN																																																																											
value	xs:string																																																																											
iname	xs:NMTOKEN																																																																											
ivalue	xs:string																																																																											
size	xs:string																																																																											
width	xs:string																																																																											
multiple	xs:boolean		false																																																																									
tabindex	xs:NMTOKEN																																																																											
id	xs:ID																																																																											
class	xs:string																																																																											
localsrc	xs:string	optional																																																																										
style	xs:NMTOKEN	optional	normal																																																																									
pivot	xs:NMTOKEN	optional																																																																										
source	<pre> <xs:element name="select"> <xs:annotation> <xs:appinfo> <comp:element-def name="select" bytecode="37"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice maxOccurs="unbounded"> <xs:element ref="wtvml:optgroup"/> <xs:element ref="wtvml:option"/> </xs:choice> <xs:attribute name="title" type="xs:string"/> <xs:attribute name="name" type="xs:NMTOKEN"/> <xs:attribute name="value" type="xs:string"/> <xs:attribute name="iname" type="xs:NMTOKEN"/> <xs:attribute name="ivalue" type="xs:string"/> <xs:attribute name="size" type="xs:string"/> <xs:attribute name="width" type="xs:string"/> <xs:attribute name="multiple" type="xs:boolean" default="false"/> <xs:attribute name="tabindex" type="xs:NMTOKEN"/> <xs:attribute name="id" type="xs:ID"/> <xs:attribute name="class" type="xs:string"/> <xs:attribute name="localsrc" type="xs:string" use="optional"/> <xs:attribute name="style" use="optional" default="normal"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="normal"/> <xs:enumeration value="autoup"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="pivot" use="optional"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl"/> <xs:enumeration value="tr"/> <xs:enumeration value="tc"/> <xs:enumeration value="ml"/> <xs:enumeration value="mc"/> <xs:enumeration value="mr"/> <xs:enumeration value="bl"/> <xs:enumeration value="bc"/> <xs:enumeration value="br"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>																																																																											

<pre> </xs:attribute> </xs:complexType> </xs:element> </pre>
--

5.5.1.1 Localsrc attribute

localsrc=CDATA

By default, items in select list will have to the "opt" icon image drawn on the screen beside them.

Application authors can replace the default image with an image of their own design, or with a blank image, using the localsrc attribute.

To remove the default icon, specify localsrc="blank", or localsrc=" " in the select element.

5.5.1.2 Style attribute

style="normal|autoup"

The style attribute specifies the behaviour of the select object. If this attribute is unspecified, or is specified as "normal" a select object will only open to reveal its select list when the user has "clicked" on the focused object.

When the style attribute is defined as an autoup, the select list will open on focus, without requiring the user to click on the object to open it.

The style attribute of the select element can be specified as part of a mode. If it is set to autoup at the mode level, then all select objects in cards referring to this mode will display the autoup behaviour.

A mode level autoup behaviour can be overridden at the card level.

5.5.1.3 Pivot attribute

pivot="(tl|tm|tr|cl|cm|cr|bl|bm|br)"

The pivot attribute specifies which corner or side of the select object will anchor the select list when it pops up.

5.5.2 Optgroup element

diagram					
namespace	http://waptv.com/xsd/wtvml				
children	wtvml:optgroup wtvml:option				
used by	elements optgroup select				
attributes	Name	Type	Use	Default	Fixed
	title	xs:string			
	id	xs:ID			
	class	xs:string			
	localsrc	xs:string	optional		
source	<pre> <xs:element name="optgroup"> <xs:annotation> <xs:appinfo> <comp:element-def name="optgroup" bytecode="34" /> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice maxOccurs="unbounded"> <xs:element ref="wtvml:optgroup" /> <xs:element ref="wtvml:option" /> </xs:choice> <xs:attribute name="title" type="xs:string" /> <xs:attribute name="id" type="xs:ID" /> <xs:attribute name="class" type="xs:string" /> <xs:attribute name="localsrc" type="xs:string" use="optional" /> </xs:complexType> </xs:element> </pre>				

5.5.2.1 Localsrc attribute

localsrc=CDATA

By default, optgroup items will have the "optg" icon image drawn on screen beside them.

Application authors can replace the default image with an image of their own design, or with a blank image, using the localsrc attribute.

To remove the default image, specify localsrc="blank", or localsrc="" in the optgroup element.

5.5.3 Option element

diagram					
namespace	http://waptv.com/xsd/wtvml				
children	wtvml:onevent				
used by	elements optgroup select				
attributes	Name	Type	Use	Default	Fixed
	value	xs:string			
	title	xs:string			
	onpick	xs:string			
	id	xs:ID			
	class	xs:string			
	localsrc	xs:string	optional		
source	<pre> <xs:element name="option"> <xs:annotation> <xs:appinfo> <comp:element-def name="option" bytecode="35"/> <comp:allowed-element name="#PCDATA"/> </xs:appinfo> </xs:annotation> <xs:complexType mixed="true"> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtvml:onevent"/> </xs:choice> <xs:attribute name="value" type="xs:string"/> <xs:attribute name="title" type="xs:string"/> <xs:attribute name="onpick" type="xs:string"/> <xs:attribute name="id" type="xs:ID"/> <xs:attribute name="class" type="xs:string"/> <xs:attribute name="localsrc" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

5.5.3.1 Localsrc attribute


localsrc=CDATA

By default, option items will have to the "opt" icon image drawn on the screen beside them.

Application authors can replace the default image with an image of their own design, or with a blank image, using the localsrc attribute.

To remove the default image, specify localsrc="blank", or localsrc="" in the option element.

5.5.4 Input element

diagram																																																																																																					
namespace	http://waptv.com/xsd/wtml																																																																																																				
children	wtml:onevent																																																																																																				
used by	elements fieldset p pre td																																																																																																				
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>xs:NMTOKEN</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>type</td> <td>xs:NMTOKEN</td> <td></td> <td>text</td> <td></td> </tr> <tr> <td>value</td> <td>xs:string</td> <td></td> <td>null</td> <td></td> </tr> <tr> <td>format</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>size</td> <td>xs:NMTOKEN</td> <td></td> <td></td> <td></td> </tr> <tr> <td>maxlength</td> <td>xs:NMTOKEN</td> <td></td> <td></td> <td></td> </tr> <tr> <td>tabindex</td> <td>xs:NMTOKEN</td> <td></td> <td></td> <td></td> </tr> <tr> <td>title</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>borderwidth</td> <td>xs:NMTOKEN</td> <td></td> <td></td> <td></td> </tr> <tr> <td>bgcolor</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>cursorcolor</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>bordercolor</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>borderunfocused</td> <td></td> <td>xs:string</td> <td></td> <td></td> </tr> <tr> <td>borderfocused</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>id</td> <td>xs:ID</td> <td></td> <td></td> <td></td> </tr> <tr> <td>class</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>label</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>localsrc</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>pos</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	name	xs:NMTOKEN	required			type	xs:NMTOKEN		text		value	xs:string		null		format	xs:string				size	xs:NMTOKEN				maxlength	xs:NMTOKEN				tabindex	xs:NMTOKEN				title	xs:string				borderwidth	xs:NMTOKEN				bgcolor	xs:string				cursorcolor	xs:string				bordercolor	xs:string				borderunfocused		xs:string			borderfocused	xs:string				id	xs:ID				class	xs:string				label	xs:string	optional			localsrc	xs:string	optional			pos	xs:string	optional		
Name	Type	Use	Default	Fixed																																																																																																	
name	xs:NMTOKEN	required																																																																																																			
type	xs:NMTOKEN		text																																																																																																		
value	xs:string		null																																																																																																		
format	xs:string																																																																																																				
size	xs:NMTOKEN																																																																																																				
maxlength	xs:NMTOKEN																																																																																																				
tabindex	xs:NMTOKEN																																																																																																				
title	xs:string																																																																																																				
borderwidth	xs:NMTOKEN																																																																																																				
bgcolor	xs:string																																																																																																				
cursorcolor	xs:string																																																																																																				
bordercolor	xs:string																																																																																																				
borderunfocused		xs:string																																																																																																			
borderfocused	xs:string																																																																																																				
id	xs:ID																																																																																																				
class	xs:string																																																																																																				
label	xs:string	optional																																																																																																			
localsrc	xs:string	optional																																																																																																			
pos	xs:string	optional																																																																																																			
source	<pre> <xs:element name="input"> <xs:annotation> <xs:appinfo> <comp:element-def name="input" bytecode="2f"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice> <xs:element ref="wtml:onevent" minOccurs="0" maxOccurs="unbounded"/> </xs:choice> <xs:attribute name="name" use="required"> <xs:annotation> <xs:documentation>The name attribute defines the name of the variable whose value is to be set with the result of the input. When used with radio and checkbox input types, the name attribute has an extended function to define logical groupings of checkbox and radio button input elements.</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"/> </xs:simpleType> </xs:attribute> <xs:attribute name="type" default="text"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="text"/> <xs:enumeration value="password"/> <xs:enumeration value="radio"/> <xs:enumeration value="checkbox"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="value" type="xs:string" default="null"/> <xs:attribute name="format" type="xs:string"/> <xs:attribute name="size" type="xs:NMTOKEN"/> <xs:attribute name="maxlength" type="xs:NMTOKEN"/> <xs:attribute name="tabindex" type="xs:NMTOKEN"/> <xs:attribute name="title" type="xs:string"/> <xs:attribute name="borderwidth" type="xs:NMTOKEN"/> <xs:attribute name="bgcolor" type="xs:string"/> <xs:attribute name="cursorcolor" type="xs:string"/> <xs:attribute name="bordercolor" type="xs:string"/> <xs:attribute name="borderunfocused" type="xs:string"/> <xs:attribute name="borderfocused" type="xs:string"/> </xs:complexType> </xs:element> </pre>																																																																																																				

<pre> <xs:attribute name="id" type="xs:ID"/> <xs:attribute name="class" type="xs:string"/> <xs:attribute name="label" type="xs:string" use="optional"/> <xs:attribute name="localsrc" type="xs:string" use="optional"/> <xs:attribute name="pos" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>

The WML 1.3 standard input attribute emptyok is an exception from the WTVML 6.3 specification.

As with WAP Forum WML, literal characters in input fields can be supported by using the "\" prefix. For example, a single field can be used to capture a date in DD/MM/YYYY format as follows:

```
<input format="DD\\DD\\DDDD"/>
```

A telephone number of the form xxxx-xxx-xxxx could be captured as follows:

```
<input format="DDDD\\-DDD\\-DDDD"/>
```

5.5.4.1 Name attribute

The name attribute defines the name of the variable whose value is to be set with the result of the input.

When used with radio and checkbox input types, the name attribute has an extended function to define logical groupings of checkbox and radio button input elements.

For example, two radio type input elements with their names set to "sex" will act as a group, and their respective values could be "male" and "female".

Two checkbox type input elements sharing a name attribute of "hobbies" could have respective values of "fishing" and "golf".

5.5.4.2 Type attribute

```
type=(text|password|checkbox|radio)
```

The type attribute is used to define the input field type. WTVML adds checkbox and radio button functionality to the standard WML input element.

When the type attribute is assigned a value of checkbox or radio, the microBrowser should render a checkbox or radio button respectively, instead of an input box. These are rendered from localsource multi-state bitmap images.

5.5.4.3 Value attribute

```
value=vdata
```

The value attribute specifies the default value for the variable to be set by the input element.

For radio and checkbox input types, the value attribute must always be defined, as the result of the user selecting a radio button or checkbox can only come from this attribute, since there will be no text input by the user.

Because this attribute is not required by text and password input types, a default value of "null" is supported.

5.5.4.4 Format attribute

```
format="(A|a|N|D|X|x|M|m)*"
```

The format attribute applies to "text" and "password" input types only. This attribute is ignored for "radio" and "checkbox" input types.

Text and password input fields can be customized to accept a variety of expected content types and to customize behaviour of the input device. The type is specified by the format attribute.

If the value of the format attribute is set to a single integer *n*, the input field is displayed as a multi-line field of *n* visible lines. Field width is determined by the size attribute. Content wraps and scrolls vertically.

If the format attribute is unspecified, the field will accept any upper case, lowercase, numeric or symbol character, and the input device behaviour will be to display lowercase letters, then uppercase letters.

For example, to input a capital "B" character in an input field with no format specified, the user would have to press the "2" button on the remote control handset six times. The input characters would cycle through "a" "b" "c" "2" and "A" before the "B" character becomes available for selection.

The lowercase "m" format is exactly the same as an unspecified format.

The uppercase "M" format is the reverse, where uppercase characters bound to each key are offered first, then the lowercase characters.

Other formats are also available to force numeric input, or upper or lowercase characters only.

Format	Description
A	"ABCDEFGHIJKLMNOPQRSTUVWXYZ" "space", "" and "-".
A	"abcdefghijklmnopqrstuvwxyz" "space", "" and "-".
N	"0123456789" "." "-" and "+".
D	"0123456789".
X	"ABCDEFGHIJKLMNOPQRSTUVWXYZ" "space".
X	"abcdefghijklmnopqrstuvwxyz" "space".
M	Uppercase text characters first, then lowercase characters, symbol and numeric characters.
M	Lowercase text characters first, then uppercase characters, symbol and numeric characters.
<i>integer</i>	If the format attribute value is specified as a single integer, the input field will be drawn as a multi-line field of <i>n</i> visible lines.

The format attribute also supports the use of modifiers as per the standard WML specification. These are as follows.

Format Modifier	Description
*	Specifies an unlimited number of characters of the selected format can be input.
N	Limits the input field to accept <i>n</i> (where <i>n</i> is from 1 to 9) characters of the specified format only.

These can be used only once in any input fields format string.

An input field designed to capture a name could be defined to accommodate one uppercase character, followed by 14 lowercase characters. This would be coded as follows:

```
<input name="firstname" maxlength="15" format="X*M" />
```

5.5.4.5 Size attribute

```
size="10"
```

The size attribute defines in pixels the width of the text input field.

The size attribute applies to "text" and "password" input types only. This attribute is ignored for "radio" and "checkbox" input types.

5.5.4.6 Maxlength attribute

```
maxlength="10"
```

The maxlength attribute defines the maximum number of characters that can be entered into the input field. Default value is an unlimited number of characters.

The maxlength attribute applies to "text" and "password" input types only. This attribute is ignored for "radio" and "checkbox" input types.

5.5.4.7 Emptyok attribute

`emptyok="true"`

In the WTVML 6.3 environment, input fields are not validated. The emptyok attribute has been excluded from the input element because the platform does not distinguish whether the field has been left empty or not.

Users can leave input fields blank if they desire.

5.5.4.8 Borderwidth attribute

`borderwidth="10"`

The borderwidth attribute defines in pixels the width of the input field border.

5.5.4.9 Bgcolor attribute

`bgcolor="#rrggbb|transp"`

The bgcolor attribute specifies in RGB values, the background colour for the input field.

5.5.4.10 Cursorcolor attribute

`cursorcolor="#rrggbb|transp"`

The cursorcolor attribute defines in RGB values the colour for the cursor in the input field

5.5.4.11 Bordercolor attribute

`bordercolor="#rrggbb|transp"`

The bordercolor attribute defines the colour of the input field border in RGB values.

5.5.4.12 Borderunfocused attribute

`borderunfocused="#rrggbb|transp"`

The borderunfocused attribute specifies in RGB values the border colour for the input field when it does not have focus.

5.5.4.13 Borderfocused attribute

`borderfocused="#rrggbb|transp"`

The borderfocused attribute specifies in RGB values the border colour for the input field when it has focus.

5.5.4.14 Label attribute

`label=%vdata`

The label attribute defines a text string to be used to label a checkbox or radio button.

This element is ignored if the input element type is set to "text" or "password".

5.5.4.15 Localsrc attribute

`localsrc=href`

The localsrc attribute defines the location of a multistate radio button or checkbox bitmap image. If no localsrc attribute is defined, the browser default localsrc image will be used.

It is possible to force the browser to not use any radio button or checkbox image by specifying `localsrc="blank"` or `localsrc=""`.

5.5.4.16 Pos attribute

pos=x,y

The pos attribute defines in pixels the x,y offset of the top left hand corner of the label text from the top left corner of the localsource checkbox or radio button image.

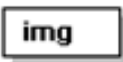
5.5.4.17 Example

```
<wml>
<card id="card1">
  <p>
    <table columns="2" colgap="10" rowgap="20">
      <tr>
        <td>
          <input name="PIN" title="Enter you 4 digit PIN" maxlength="4" size="5" format="DDDD"
type="text"/>
        </td>
        <td>
          <input name="Password" title="Enter your password" maxlength="8" size="9" format="8m"
type="password"/>
        </td>
      </tr>

      <tr>
        <td>
          <do type="vnd.wtv-g" label="Confirm">
            <go href="http://xyz.org/xxzz" />
          </do>
        </td>
        <td>
          <do type="vnd.wtv-r" label="Cancel">
            <go href="http://xyz.org/abc" />
          </do>
        </td>
      </tr>
    </table>
  </p>
</card>
</wml>
```

5.6 Images

5.6.1 Img element

diagram					
namespace	http://waptv.com/xsd/wtvml				
used by	elements a anchor b big em fieldset font i p small strong td u				
attributes	Name	Type	Use	Default	Fixed
	alt	xs:string			
	src	xs:string			
	localsrc	xs:string			
	vspace	xs:string		0	
	hspace	xs:string		0	
	align	xs:NMTOKEN		bottom	
	height	xs:string			
	width	xs:string			
	frames	xs:NMTOKEN			
	loop	xs:NMTOKEN			
	ticks	xs:NMTOKEN			
	index	xs:NMTOKEN			
	id	xs:ID			
	class	xs:string			
	cache-control	xs:NMTOKEN		cache	
source	<pre> <xs:element name="img"> <xs:annotation> <xs:appinfo> <comp:element-def name="img" bytecode="2e"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="alt" type="xs:string"/> <xs:attribute name="src" type="xs:string"/> <xs:attribute name="localsrc" type="xs:string"/> <xs:attribute name="vspace" type="xs:string" default="0"/> <xs:attribute name="hspace" type="xs:string" default="0"/> <xs:attribute name="align" default="bottom"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="top"/> <xs:enumeration value="middle"/> <xs:enumeration value="bottom"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="height" type="xs:string"/> <xs:attribute name="width" type="xs:string"/> <xs:attribute name="frames" type="xs:NMTOKEN"/> <xs:attribute name="loop" type="xs:NMTOKEN"/> <xs:attribute name="ticks" type="xs:NMTOKEN"/> <xs:attribute name="index" type="xs:NMTOKEN"/> <xs:attribute name="id" type="xs:ID"/> <xs:attribute name="class" type="xs:string"/> <xs:attribute name="cache-control" default="cache"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="cache"/> <xs:enumeration value="nocache"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>				

Additional attributes have been added to the img element to support complex behaviours and properties for animated bitmap images.

5.6.1.1 Frames attribute

frames="10"

The frames attribute is a read only attribute, and cannot be set from the WML. The frames attribute can be read from a getattr statement and displays what frame number is currently displayed from a multiframe image.

5.6.1.2 Loop attribute

loop="10"

The loop attribute specifies the repeat behaviour of a looping animated bitmap image. If unspecified, the default behaviour is to loop continuously.

Value	Description
-1	Repeat continuously.
0	Stop looping.
>1	Loop n times.

5.6.1.3 Ticks attribute

ticks="10"

The ticks attribute specifies the frequency at which frames of an animated bitmap image should be displayed.

The ticks attribute is measured in tenths of a second. An animated bitmap image with a tick attribute of "10" will animate every 1 second.

If unspecified, the default behaviour is to animate on every tick, which is every tenth of a second.

5.6.1.4 Index attribute

index="10"

The index attribute is used to set or get the current frame index.

5.6.1.5 Cache-control attribute

cache-control="(cache|nocache)"

The cache-control attribute specifies how the browser should cache this resource. There are two possible values for the cache-control attribute. The default value is "cache".

Value	Description
Cache	Keep the resource in cache until the memory is required by the browser.
Nocache	Do not cache the resource.

5.7 Tables

5.7.1 Table element

diagram	<p>The WML table element 1..∞ table row</p>				
namespace	http://waptv.com/xsd/wtml				
children	wtml:tr				
used by	elements				
	b big em fieldset font i p small strong td u				
attributes	Name	Type	Use	Default	Fixed
	align	xs:string			
	columns	xs:NMTOKEN			
	rowgap	xs:NMTOKEN			
	colgap	xs:NMTOKEN			
	vspace	xs:NMTOKEN			
	hspace	xs:NMTOKEN			
	hpad	xs:NMTOKEN			
	vpad	xs:NMTOKEN			
	colwidths	xs:string			
	bgimage	xs:string			
	bgcolor	xs:string			
	bgradius	xs:NMTOKEN			
	bordercolor	xs:string			
	borderwidth	xs:NMTOKEN			
	id	xs:ID			
	class	xs:string			
	update	xs:NMTOKEN	optional	auto	
source	<pre> <xs:element name="table"> <xs:annotation> <xs:documentation>The WML table element</xs:documentation> <xs:appinfo> <comp:element-def name="table" bytecode="1f"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:sequence maxOccurs="unbounded"> <xs:element ref="wtml:tr"/> </xs:sequence> <xs:attribute name="align" type="xs:string"/> <xs:attribute name="columns" type="xs:NMTOKEN"/> <xs:attribute name="rowgap" type="xs:NMTOKEN"/> <xs:attribute name="colgap" type="xs:NMTOKEN"/> <xs:attribute name="vspace" type="xs:NMTOKEN"/> <xs:attribute name="hspace" type="xs:NMTOKEN"/> <xs:attribute name="hpad" type="xs:NMTOKEN"/> <xs:attribute name="vpad" type="xs:NMTOKEN"/> <xs:attribute name="colwidths" type="xs:string"/> <xs:attribute name="bgimage" type="xs:string"/> <xs:attribute name="bgcolor" type="xs:string"/> <xs:attribute name="bgradius" type="xs:NMTOKEN"/> <xs:attribute name="bordercolor" type="xs:string"/> <xs:attribute name="borderwidth" type="xs:NMTOKEN"/> <xs:attribute name="id" type="xs:ID"/> <xs:attribute name="class" type="xs:string"/> <xs:attribute name="update" use="optional" default="auto"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="auto"/> <xs:enumeration value="oncardnav"/> <xs:enumeration value="noauto"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>				

Extensive modification has been made to the standard WML 1.3 table to support rich formatting of WTVML tables.

It is also possible to control layout by inserting tables within tables.

In version 5.0.x of the browser, any table definitions defined in the main Browser should also propagate into any subsidiary browsers.

5.7.1.1 Align attribute

`align=(L|C|R|D)`

The align attribute defines alignment of text and images within the columns of the table.

Format code	Description
L	Left aligned.
C	Centre aligned.
R	Right aligned.
D	Default aligned (same as Left aligned).

5.7.1.2 Columns attribute

`columns="2"`

The columns attribute specifies the number of columns to be drawn in the table. The number of columns must not be set to zero.

5.7.1.3 Rowgap attribute

`rowgap="10"`

The rowgap attribute defines the number of pixels of vertical space to be placed between each row in the table.

5.7.1.4 Colgap attribute

`colgap="10"`

The colgap attribute defines in pixels the horizontal space to be inserted between each column.

5.7.1.5 Vspace attribute

`vspace="10"`

The vspace attribute specifies in pixels the vertical space to be left above and below the table.

5.7.1.6 Hspace attribute

`hspace="10"`

The hspace attribute specifies in pixels the horizontal space to be left either side of the table.

5.7.1.7 Hpad attribute

`hpad="10"`

The hpad attribute defines in pixels the horizontal space to be left between the border of the table's cells, and the contents of the cells.

5.7.1.8 Vpad attribute

`vpad="10"`

The vpad attribute defines in pixels the vertical space to be left between the border of the table's cells, and the contents of the cells.

5.7.1.9 Colwidths attribute

colwidths=%vdata

The colwidths attribute is used to define in pixels individually customized widths for columns in a table.

The colwidths attribute can contain a series of comma separated values, with the first value defining the width of the left most column in pixels or in percentages of table width.

- `<table columns="5" colwidths="10,15,10,20,70">`

The "*" character can also be used to tell the browser to use all left over space for this column

- `<table columns="5" colwidths="10,15,10,20,*">`

To optimize performance, application authors are advised to always define column widths using numeric pixel definitions. This allows the browser to layout the table without having to perform the time consuming calculations required to determine column widths dynamically.

5.7.1.10 Bgimage attribute

bgimage="HREF"

The bgimage attribute specifies the location of an MPEG image to be displayed as the background image in the browser.

5.7.1.11 Bgcolor attribute

bgcolor="(#rrggb|transp)"

The bgcolor attribute specifies in RGB values, the background colour for the browser.

5.7.1.12 Bgradius attribute

bgradius="10"

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of table cells.

5.7.1.13 Bordercolor attribute

bordercolor="(#rrggb|transp)"

The bordercolor attribute defines the colour of the table border in RGB values.

5.7.1.14 Borderwidth attribute

borderwidth="10"

The borderwidth attribute defines in pixels the width of the table border.

5.7.1.15 Update attribute

update=(auto | oncardnav | noauto)

The update attribute controls the update behaviour for content in a dynamic broadcast content churn.

Value	Description
Auto	Browser should perform an automatic screen refresh as soon as it detects new content in the broadcast stream. This means displayed can change without any user interaction. This is the default value for the update attribute.
Oncardnav	Browser should refresh content when the browser is navigated to another card, including cards in the same deck.
Noauto	Browser should only refresh when it is navigated to a new deck.

Default value for update is "auto".

5.7.1.16 Example

```

<wml>

<card id="card1" title="Card1" paragap="10">
<p>
  <table bgcolor="#10105A" bgradius="20" hpad="5" vpad="5" bordercolor="#xxxxxx" borderwidth="3">

    <tr>
      <td hpad="5">
        <b>Select the Product from below</b>
      </td>
    </tr>

  </table>
</p>

<p>
<table rowgap='10'>

  <tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
    <onevent type='onactivate'>
      <go href="http://xyz.org/xx" />
    </onevent>
    <td width='400' hspace='20'>Notepad
    </td>
  </tr>

  <tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
    <onevent type='onactivate'>
      <go href="http://xyz.org/xx2" />
    </onevent>
    <td width='400' hspace='20'>Pocket PC
    </td>
  </tr>

  <tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
    <onevent type='onactivate'>
      <go href="http://xyz.org/xx3" />
    </onevent>
    <td width='400' hspace='20'>Desktop
    </td>
  </tr>

  <tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
    <onevent type='onactivate'>
      <go href="http://xyz.org/xx4" />
    </onevent>
    <td width='400' hspace='20'>PDA
    </td>
  </tr>

</table>
</p>
</card>
</wml>

```

5.7.2 Tr element

Diagram						
Namespace	http://waptv.com/xsd/wtvml					
Children	wtvml:td wtvml:onevent					
used by	element table					
Attributes	Name	Type	Use	Default	Fixed	
	title	xs:string				
	height	xs:NMTOKEN				
	localsrc	xs:string				
	bgimage	xs:string				
	bgradius	xs:NMTOKEN				
	bgcolor	xs:string				
	bgfocused	xs:string				
	bgactivated	xs:string				
	borderwidth	xs:NMTOKEN				
	bordercolor	xs:string				
	borderfocused	xs:string				
	borderactivated	xs:string				
	id	xs:ID				
	class	xs:string				
	update	xs:NMTOKEN	optional	auto		
	tabindex	xs:integer	optional			
source	<pre> <xs:element name="tr"> <xs:annotation> <xs:documentation>table row</xs:documentation> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtvml:td" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:onevent" minOccurs="0"/> </xs:choice> <xs:attribute name="title" type="xs:string"/> <xs:attribute name="height" type="xs:NMTOKEN"/> <xs:attribute name="localsrc" type="xs:string"/> <xs:attribute name="bgimage" type="xs:string"/> <xs:attribute name="bgradius" type="xs:NMTOKEN"/> <xs:attribute name="bgcolor" type="xs:string"/> <xs:attribute name="bgfocused" type="xs:string"/> <xs:attribute name="bgactivated" type="xs:string"/> <xs:attribute name="borderwidth" type="xs:NMTOKEN"/> <xs:attribute name="bordercolor" type="xs:string"/> <xs:attribute name="borderfocused" type="xs:string"/> <xs:attribute name="borderactivated" type="xs:string"/> <xs:attribute name="id" type="xs:ID"/> <xs:attribute name="class" type="xs:string"/> <xs:attribute name="update" use="optional" default="auto"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="auto"/> <xs:enumeration value="oncardnav"/> <xs:enumeration value="noauto"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="tabindex" type="xs:integer" use="optional"/> </xs:complexType> </xs:element> </pre>					

The tr element is used as a container to hold a single table row. The tr element has been extensively modified to allow rich design of table rows, and to accommodate onevent triggering.

5.7.2.1 Title attribute

`title=%vdata`

The title attribute can be used to display help information concerning this table row. The value of the `title` attribute is displayed in the Title pane when this table row has focus.

5.7.2.2 Height attribute

`height="10"`

The height attribute specified in pixels the height of the table row.

5.7.2.3 Localsrc attribute

`localsrc=%vdata`

The `localsrc` attribute defines the local source image to be associated with this table row.

5.7.2.4 Bgimage attribute

`bgimage=href`

The `bgimage` attribute specifies the location of the background image to be used in this table row.

5.7.2.5 Bgradius attribute

`bgradius="10"`

The `bgradius` attribute defines in pixels the radius of a curve used to "round off" the corners of table cells.

5.7.2.6 Bgcolor attributes

`bgcolor="#rrggbb|transp"`

`bgfocused="#rrggbb|transp"`

`bgactivated="#rrggbb|transp"`

By specifying different background colours for each of the three possible states for table rows: focused, unfocused and activated; it is possible to provide useful navigation feedback.

5.7.2.7 Borderwidth attribute

`borderwidth="10"`

The `borderwidth` attribute defines in pixels the width of the table row border.

5.7.2.8 Bordercolor attributes

`bordercolor="#rrggbb|transp"`

`borderfocused="#rrggbb|transp"`

`borderactivated="#rrggbb|transp"`

By specifying different border colours for each of the three possible states for table rows: focused, unfocused and activated; it is possible to provide useful navigation feedback.

5.7.2.9 Update attribute

update=(auto | oncardnav | noauto)

The update attribute controls the update behaviour for content in a dynamic broadcast content churn.

Value	Description
Auto	Browser should perform an automatic screen refresh as soon as it detects new content in the broadcast stream. This means displayed can change without any user interaction. This is the default value for the update attribute.
Oncardnav	Browser should refresh content when the browser is navigated to another card, including cards in the same deck.
Noauto	Browser should only refresh when it is navigated to a new deck.

Default value for update is "auto".

5.7.2.10 Example

```
<wml>

<card id="card1" title="Card1" paragap="10">
<p>
  <table bgcolor="#10105A" bgradius="20" hpad="5" vpad="5" bordercolor="#xxxxxx" borderwidth="3">

    <tr>
      <td hpad="5">
        <b>Select the Product from below</b>
      </td>
    </tr>

  </table>
</p>

<p>
<table rowgap='10'>

  <tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
    <onevent type='onactivate'>
      <go href="http://xyz.org/xx" />
    </onevent>
    <td width='400' hspace='20'>Notepad
    </td>
  </tr>

  <tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
    <onevent type='onactivate'>
      <go href="http://xyz.org/xx2" />
    </onevent>
    <td width='400' hspace='20'>Pocket PC
    </td>
  </tr>

  <tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
    <onevent type='onactivate'>
      <go href="http://xyz.org/xx3" />
    </onevent>
    <td width='400' hspace='20'>Desktop
    </td>
  </tr>

  <tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
    <onevent type='onactivate'>
      <go href="http://xyz.org/xx4" />
    </onevent>
    <td width='400' hspace='20'>PDA
    </td>
  </tr>

</table>

</table>
```


	bgfocused xs:string bgactivated xs:string borderwidth xs:NMTOKEN bordercolor xs:string borderfocused xs:string borderactivated xs:string id xs:ID class xs:string mode xs:NMTOKEN wrap update xs:NMTOKEN optional auto tabindex xs:integer optional
annotation	appInfo <comp:element-def name="td" bytecode="1d"/> <comp:allowed-element name="#PCDATA"/>
source	<pre> <xs:element name="td"> <xs:annotation> <xs:appinfo> <comp:element-def name="td" bytecode="1d"/> <comp:allowed-element name="#PCDATA"/> </xs:appinfo> </xs:annotation> <xs:complexType mixed="true"> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtml:em"/> <xs:element ref="wtml:strong"/> <xs:element ref="wtml:b"/> <xs:element ref="wtml:i"/> <xs:element ref="wtml:u"/> <xs:element ref="wtml:big"/> <xs:element ref="wtml:small"/> <xs:element ref="wtml:font"/> <xs:element ref="wtml:br"/> <xs:element ref="wtml:img"/> <xs:element ref="wtml:anchor"/> <xs:element ref="wtml:a"/> <xs:element ref="wtml:table"/> <xs:element ref="wtml:input"/> <xs:element ref="wtml:select"/> <xs:element ref="wtml:fieldset"/> <xs:element ref="wtml:do"/> <xs:element ref="wtml:onevent"/> </xs:choice> <xs:attribute name="align" type="xs:string"/> <xs:attribute name="title" type="xs:string"/> <xs:attribute name="height" type="xs:NMTOKEN"/> <xs:attribute name="width" type="xs:NMTOKEN"/> <xs:attribute name="hspace" type="xs:NMTOKEN"/> <xs:attribute name="vspace" type="xs:NMTOKEN"/> <xs:attribute name="hpad" type="xs:NMTOKEN"/> <xs:attribute name="vpad" type="xs:NMTOKEN"/> <xs:attribute name="linegap" type="xs:NMTOKEN"/> <xs:attribute name="colspan" type="xs:NMTOKEN" default="1"/> <xs:attribute name="rowspan" type="xs:NMTOKEN" default="1"/> <xs:attribute name="localsrc" type="xs:string"/> <xs:attribute name="bgimage" type="xs:string"/> <xs:attribute name="bgradius" type="xs:NMTOKEN"/> <xs:attribute name="bgcolor" type="xs:string"/> <xs:attribute name="bgfocused" type="xs:string"/> <xs:attribute name="bgactivated" type="xs:string"/> <xs:attribute name="borderwidth" type="xs:NMTOKEN"/> <xs:attribute name="bordercolor" type="xs:string"/> <xs:attribute name="borderfocused" type="xs:string"/> <xs:attribute name="borderactivated" type="xs:string"/> <xs:attribute name="id" type="xs:ID"/> <xs:attribute name="class" type="xs:string"/> <xs:attribute name="mode" default="wrap"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="wrap"/> <xs:enumeration value="nowrap"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="update" use="optional" default="auto"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> </pre>

```

        <xs:enumeration value="auto" />
        <xs:enumeration value="oncardnav" />
        <xs:enumeration value="noauto" />
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="tabindex" type="xs:integer" use="optional"/>
</xs:complexType>
</xs:element>

```

The td element is used as a container to hold a single table cell data within a table row. The td element has been extensively modified to contain any object, including anchors, dos, sub-tables and input fields. This makes it possible to lay out all these object types using tables.

Formatting definitions (such as background colours, border colours, border radius) for a child td elements will overwrite any formatting applied to its tr parent elements.

5.7.3.1 Align attribute

align=(L|C|R|D)

The align attribute defines alignment of text and images within the columns of the table.

Format code	Description
L	Left aligned.
C	Centre aligned.
R	Right aligned.
D	Default aligned (same as Left aligned).

5.7.3.2 Title attribute

title=%vdata

The title attribute can be used to display help information concerning this table cell. The value of the title attribute is displayed in the Title pane when this table cell has focus.

5.7.3.3 Height attribute

height="10"

The height attribute specified in pixels the height of the table cell.

5.7.3.4 Width attribute

width="10"

The width attribute specified in pixels the width of the table cell.

5.7.3.5 Hspace attribute

hspace="10"

The hspace attribute specifies in pixels the horizontal space to be left either side of the table cell.

5.7.3.6 Vspace attribute

vspace="10"

The vspace attribute specifies in pixels the vertical space to be left above and below the table cell.

5.7.3.7 Hpad attribute

`hpad="10"`

The `hpad` attribute defines in pixels the horizontal space to be left between the border of the table cell, and the contents of the cell.

5.7.3.8 Vpad attribute

`vpad="10"`

The `vpad` attribute defines in pixels the vertical space to be left between the border of the table cell, and the contents of the cell.

5.7.3.9 Linegap attribute

`linegap="10"`

The `linegap` attribute specifies in pixels the amount of space to be left between each line of text in the table cell.

5.7.3.10 Colspan attribute

The `colspan` attribute defines the number of table columns spanned by the cell.

5.7.3.11 Rowspan attribute

The `rowspan` attribute defines the number of table rows to be spanned by the cell.

5.7.3.12 Localsrc attribute

`localsrc=%vdata`

The `localsrc` attribute defines the local source image to be associated with this table cell.

5.7.3.13 Bgimage attribute

`bgimage=href`

The `bgimage` attribute specifies the location of the background image to be used in this table cell.

5.7.3.14 Bgradius attribute

`bgradius="10"`

The `bgradius` attribute defines in pixels the radius of a curve used to "round off" the corners of the table cell.

5.7.3.15 Bgcolor attributes

`bgcolor="#rrggbb|transp"`

`bgfocused="#rrggbb|transp"`

`bgactivated="#rrggbb|transp"`

By specifying different background colours for each of the three possible states for table cells: focused, unfocused and activated; it is possible to provide useful navigation feedback.

5.7.3.16 Borderwidth attribute

borderwidth="10"

The borderwidth attribute defines in pixels the width of the table cell border.

5.7.3.17 Bordercolor attributes

bordercolor="(#rrggb|transp)"

borderfocused="(#rrggb|transp)"

borderactivated="(#rrggb|transp)"

By specifying different border colours for each of the three possible states for table cells: focused, unfocused and activated; it is possible to provide useful navigation feedback.

5.7.3.18 Mode attribute

mode=(wrap | nowrap)

The mode attribute defines whether or not content should wrap within the table cell. Default value is "wrap".

5.7.3.19 Update attribute

update=(auto | oncardnav | noauto)

The update attribute controls the update behaviour for content in a dynamic broadcast content churn.

Value	Description
Auto	Browser should perform an automatic screen refresh as soon as it detects new content in the broadcast stream. This means displayed can change without any user interaction. This is the default value for the update attribute.
oncardnav	Browser should refresh content when the browser is navigated to another card, including cards in the same deck.
Noauto	Browser should only refresh when it is navigated to a new deck.

Default value for update is "auto".

5.7.3.20 Example

```
<wml>

<card id="card1" title="Card1" paragap="10">
<p>
  <table bgcolor="#10105A" bgradius="20" hpad="5" vpad="5" bordercolor="#xxxxxx" borderwidth="3">

    <tr>
      <td hpad="5">
        <b>Select the Product from below</b>
      </td>
    </tr>

  </table>
</p>

<p>
<table rowgap='10'>

  <tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
    <onevent type='onactivate'>
      <go href="http://xyz.org/xx" />
    </onevent>
    <td width='400' hspace='20'>Notepad
  </td>
</tr>
```

```
<tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
  <onevent type='onactivate'>
    <go href="http://xyz.org/xx2" />
  </onevent>
  <td width='400' hspace='20'>Pocket PC
</td>
</tr>

<tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
  <onevent type='onactivate'>
    <go href="http://xyz.org/xx3" />
  </onevent>
  <td width='400' hspace='20'>Desktop
</td>
</tr>

<tr bgcolor='#928FAE' bgfocused='#10105A' bgactivated='#57518F' title='Use the arrow keys, then
press Select'>
  <onevent type='onactivate'>
    <go href="http://xyz.org/xx4" />
  </onevent>
  <td width='400' hspace='20'>PDA
</td>
</tr>

</table>
</p>
</card>
</wml>
```

5.8 Text

5.8.1 Font element

diagram																																									
namespace	http://waptv.com/xsd/wtvml																																								
children	wtvml:br wtvml:anchor wtvml:a wtvml:table wtvml:input wtvml:do wtvml:select wtvml:img																																								
used by	elements fieldset p td																																								
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>xs:NMTOKEN</td> <td></td> <td></td> <td></td> </tr> <tr> <td>fgstatic</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>fgcolor</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>fgfocused</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>fgactivated</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> <tr> <td>id</td> <td>xs:ID</td> <td></td> <td></td> <td></td> </tr> <tr> <td>class</td> <td>xs:string</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	size	xs:NMTOKEN				fgstatic	xs:string				fgcolor	xs:string				fgfocused	xs:string				fgactivated	xs:string				id	xs:ID				class	xs:string			
Name	Type	Use	Default	Fixed																																					
size	xs:NMTOKEN																																								
fgstatic	xs:string																																								
fgcolor	xs:string																																								
fgfocused	xs:string																																								
fgactivated	xs:string																																								
id	xs:ID																																								
class	xs:string																																								
source	<pre> <xs:element name="font"> <xs:annotation> <xs:appinfo> <comp:element-def name="font" bytecode="16"/> <comp:allowed-element name="#PCDATA"/> </xs:appinfo> </xs:annotation> <xs:complexType mixed="true"> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtvml:br" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:anchor" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:a" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:table" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:input" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:do" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:select" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:img" minOccurs="0" maxOccurs="unbounded"/> </xs:choice> <xs:attribute name="size" type="xs:NMTOKEN"/> <xs:attribute name="fgstatic" type="xs:string"/> <xs:attribute name="fgcolor" type="xs:string"/> <xs:attribute name="fgfocused" type="xs:string"/> <xs:attribute name="fgactivated" type="xs:string"/> <xs:attribute name="id" type="xs:ID"/> <xs:attribute name="class" type="xs:string"/> </xs:complexType> </xs:element> </pre>																																								

The font element is a new element introduced in WTVML. The standard WAP Forum WML 1.3 spec does not include a font element.

The font element specifies the size and colour to be applied to fonts in WTVML. This element can be used to override the font attributes defined in the current fontset.

Fonts can be vector fonts, or bitmap graphic files, with either 1 bit or 2 bit colour.

The application author can assign colours to the font to determine its colour. The background colour against which the font will be displayed is determined by the context of the WML in which the font is used. This is not specified within the font element.

If the font being used is a 2 bit colour bitmap, the browser should automatically calculate and use antialias colours based on the current foreground colour and the background colour.

Antialiasing is not possible with 1 bit and vector font resources.

NOTE: Font elements must always be identified by an id attribute. The id attribute is part of the WML coreattrs. The default WML text emphasis formats (strong, i etc) can be used as id if desired. It is also possible to use more descriptive names such as Helvetica or Times.

5.8.1.1 Size attribute

size="10"

The size attribute is used to scale vector fonts. The size attribute defines in pixels the vertical height of the font. The browser ignores this attribute if the font is a bitmap font.

5.8.1.2 Fgstatic attribute

fgstatic=CDATA

The fgstatic attribute specifies the foreground colour to be applied to all static text. Static text refers to any text element that is not an anchor or a do.

5.8.1.3 Fgcolor attribute

fgcolor=CDATA

The fgcolor attribute specifies the foreground colour to be applied to a "clickable" text element. A WTVML microbrowser does not automatically apply an underline to actionable text elements. The application author should define a distinguishing foreground colour to be applied to actionable text elements.

5.8.1.4 Fgfocused attribute

fgfocused=CDATA

The fgfocused attribute specifies the font colour when the actionable text element has focus.

5.8.1.5 Fgactivated attribute

fgactivated=CDATA

The fgactivated attribute specifies the font colour when the actionable text element is activated.

5.8.1.6 Example

```
<font id="helvetica" size="20" fgstatic="#ffffff" fgcolor="#cccccc" fgfocused="#f00f00"
fgactivated="#f0f0f0">
SAMPLE
</font>
```

5.8.2 Paragraphs

diagram	
namespace	http://waptv.com/xsd/wtvml
children	wtvml:em wtvml:strong wtvml:b wtvml:i wtvml:u wtvml:big wtvml:small wtvml:font wtvml:br wtvml:img wtvml:anchor wtvml:a wtvml:table wtvml:input wtvml:select wtvml:fieldset wtvml:do
used by	element card

attributes	Name	Type	Use	Default	Fixed
	align	xs:NMTOKEN	optional	left	
	mode	xs:NMTOKEN	optional		
	linegap	xs:NMTOKEN	optional		
	id	xs:ID	optional		
	class	xs:string	optional		
	update	xs:NMTOKEN	optional	auto	
annotation	appInfo <comp:element-def name="p" bytecode="20"/> <comp:allowed-element name="#PCDATA"/>				
source	<pre> <xs:element name="p"> <xs:annotation> <xs:appinfo> <comp:element-def name="p" bytecode="20"/> <comp:allowed-element name="#PCDATA"/> </xs:appinfo> </xs:annotation> <xs:complexType mixed="true"> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="wtvml:em" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:strong" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:b" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:i" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:u" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:big" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:small" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:font" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:br" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:img" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:anchor" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:a" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:table" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:input" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:select" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:fieldset" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="wtvml:do" minOccurs="0" maxOccurs="unbounded"/> </xs:choice> <xs:attribute name="align" use="optional" default="left"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="left"/> <xs:enumeration value="right"/> <xs:enumeration value="center"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="mode" use="optional"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="wrap"/> <xs:enumeration value="nowrap"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="linegap" type="xs:NMTOKEN" use="optional"/> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> <xs:attribute name="update" use="optional" default="auto"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="auto"/> <xs:enumeration value="oncardnav"/> <xs:enumeration value="noauto"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>				

The paragraph element has been extended with the addition of the linegap and update attributes.

5.8.2.1 Linegap attribute

linegap="10"

The linegap attribute defines in pixels the space that should be placed between lines within the paragraph.

5.8.2.2 Update attribute

update=(auto | oncardnav | noauto)

The update attribute controls the update behaviour for content in a dynamic broadcast content churn.

Value	Description
auto	Browser should perform an automatic screen refresh as soon as it detects new content in the broadcast stream. This means displayed can change without any user interaction. This is the default value for the update attribute.
oncardnav	Browser should refresh content when the browser is navigated to another card, including cards in the same deck.
noauto	Browser should only refresh when it is navigated to a new deck.

Default value for update is "auto".

5.8.2.3 Example

```
<?xml version="1.0"?>

<wml>

<head>
  <meta name="WTV_Connect_Now" content="" />
</head>

<template>
  <do type="prev">
    <go href='home.wml' />
  </do>

  <do type='help' label=''>
    <go href='indexhelp.wml' />
  </do>
</template>

<card id="login" title='Welcome'>

  <do type='vnd.wtv-r' label='T&Cs' title="Press Select to read Terms & Conditions">
    <go href='terms.wml' />
  </do>

<p linegap="4">

  SAMPLE

</p>

</card>
</wml>
```

6 WTVML 6.3 Modeset specification

6.1 Introduction

A modeset file defines one or more modes which in turn define the characteristics and location of the WTVML platform onscreen objects.

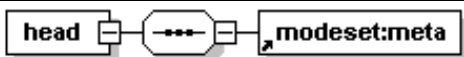
Modeset files are XML files, structured according to the rules defined in modeset.dtd. A modeset XML file can have any name and can live at any point of the wtvml application webserver hierarchy.

Modeset XML files are commonly called modeset.xml and are found in the /modeset/ subdirectory of the web hierarchy.

6.2 Cache control


Cache behaviour for modeset files can be controlled through the meta element in the head of the modesetinstance.xml file, in the same way that cache behaviour for wtvml files can be defined in the meta element.

6.2.1 Head element

diagram	
namespace	http://waptv.com/xsd/modeset
children	modeset:meta
used by	element modeset
source	<pre><xs:element name="head"> <xs:complexType> <xs:sequence> <xs:element ref="modeset:meta"/> </xs:sequence> </xs:complexType> </xs:element></pre>

The head element can contain a single meta element, and has no attributes.

6.2.2 Meta element

diagram																			
namespace	http://waptv.com/xsd/modeset																		
used by	element head																		
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>http-equiv</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>content</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	http-equiv	xs:string	optional				content	xs:string	optional			
Name	Type	Use	Default	Fixed	Annotation														
http-equiv	xs:string	optional																	
content	xs:string	optional																	
source	<pre><xs:element name="meta"> <xs:complexType> <xs:attribute name="http-equiv" type="xs:string" use="optional"/> <xs:attribute name="content" type="xs:string" use="optional"/> </xs:complexType> </xs:element></pre>																		

6.2.2.1 Http-equiv attribute

The http-equiv attribute provides a mechanism to set an http header, where the http-equiv value is the header name, and content is the header value. The network's transcoding server will convert an "http-equiv" meta value to an http response header before sending the file to the WTVML microbrowser.


6.2.2.2 Content attribute

When set to a value of "no-cache", the http-equiv meta value will instruct the microbrowser not to cache the modesetinstance.xml file.

This allows application authors to navigate away from a development service, then re-enter the service to view any modeset changes, rather than having to exit the browser completely before being able to view modifications to modesets.

6.3 WTVML 6.3 Screen objects

6.3.1 AV element

diagram						
namespace	http://waptv.com/xsd/modeset					
used by	element					
attributes	Name	Type	Use	Default	Fixed	Annotation
	id	xs:string	optional	av		
	mode	xs:NMTOKEN		optional	blank	
	rect-src	xs:string	optional			
	rect	xs:string	optional			
	hscale	xs:string	optional	1.0		
	vscale	xs:string	optional	1.0		
	video	xs:string	optional			
	audio	xs:string	optional			
	toggle	xs:string	optional			
	offset	xs:string	optional			
	bgimage	xs:string	optional			
	channel	xs:string	optional			
	skey	xs:string	optional			
annotation	documentation Defines characteristics of the AV object for display of broadcast audio and video content. applInfo <comp:element-def name="av" bytecode="07"/>					
source	<pre><xs:element name="av"> <xs:annotation> <xs:documentation>Defines characteristics of the AV object for display of broadcast audio and video content.</xs:documentation> <xs:appinfo> <comp:element-def name="av" bytecode="07"/> </xs:appinfo> </xs:annotation> </xs:element> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional" default="av"/> <xs:attribute name="mode" use="optional" default="blank"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="full"/> <xs:enumeration value="window"/> <xs:enumeration value="blank"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="rect-src" type="xs:string" use="optional"/> <xs:attribute name="rect" type="xs:string" use="optional"/> <xs:attribute name="hscale" type="xs:string" use="optional" default="1.0"/> <xs:attribute name="vscale" type="xs:string" use="optional" default="1.0"/> <xs:attribute name="video" type="xs:string" use="optional"/> <xs:attribute name="audio" type="xs:string" use="optional"/> <xs:attribute name="toggle" type="xs:string" use="optional"/> <xs:attribute name="offset" type="xs:string" use="optional"/> <xs:attribute name="bgimage" type="xs:string" use="optional"/> <xs:attribute name="service" type="xs:string" use="optional"/> </xs:complexType> </xs:element></pre>					

<pre></xs:complexType> </xs:element></pre>
--

The av element defines the configuration for the Audio-Video object in the browser.

The av element controls:

- whether video should be full screen, quarter screen or blank;
- positioning of the video window;
- scaling of the video;
- the location of the audio or video resource to be displayed;
- the key that will toggle between the browser and full screen video.

The av element exposes certain scaling functionality of the set-top-box through the WML. Source video can be scaled horizontally to any degree.

6.3.1.1 Id attribute

id=av

There should be only a single video instance defined, and it should be allocated the id av. This attribute can be manipulated from the WML content using the setattr element.

6.3.1.2 Mode attribute

mode=(full|window|blank)

This attribute specifies how the video resource should be displayed:

- full - video resource is displayed in full screen mode;
- window - video resource is displayed in a pane within the browser window;
- blank - no video is displayed.

6.3.1.3 Rect-src attribute

rect-src="x1,y1,x2,y2"

The rect-src attribute provides an optional method to define a subframe of a video stream source frame to be displayed in the AV object widow. The rect-src attribute is defined relative to the video source frame, and has no bearing on where the AV window will be positioned onscreen. This is defined in the rect attribute.

This is useful where there is a requirement to selectively display areas of a matrix-style video source.

Where rect-src is used, any scale or offset attribute values will be ignored.

```
<av mode='window' rect-src='0,144,176,288' rect='320,32,672,320' video='track://VID1?hsize=0.48&vsize=0.5' />
```

In this example, the source video frame has a resolution of 352×288 pixels (determined by the 0.48 : 0.5 ratio defined against full screenvideo of 720×576 pixels), and the rect-src attribute is selecting the lower left-hand quarter rectangle in this frame.

Note that the AV object rect dimension does not need to be the same size or aspect ratio as the rect-src dimension. This is important as it allows the original video source matrix to be aspect distorted for later aspect correction in the AV window.

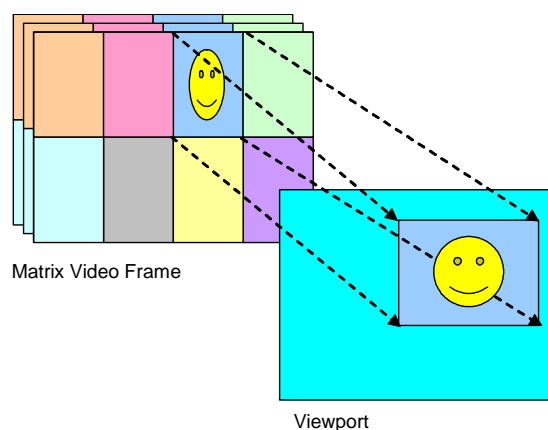


Figure 2: Aspect ratio distortion and correction

6.3.1.4 Rect attribute

`rect=x1,y1,x2,y2`

The rect attribute defines where the av window is to be drawn on the screen.

6.3.1.5 Hscale attribute

`hscale="10"`

The hscale attribute determines horizontal scaling for the video window defined by the av element. The hscale attribute is always defined relative to full screen video, and can be an arbitrary value.

6.3.1.6 Vscale attribute

`vscale=(0.5 | 1.0 | 2.0)`

The vscale attribute determines vertical scaling for the video window defined by the av element. The vscale attribute is always defined relative to full screen video. The valid values for vscale are 0,5, 1,0 or 2,0. That is, the vscale attribute is limited to only halving (by removing every other line from the broadcast) or doubling (by doubling every line) the vertical scale of the source video.

Because of this, application authors are effectively limited to scaling the x and y axes of the source video by $0,5 \times 0,5$ or 2×2 .

Although arbitrary scaling on the horizontal axis is possible, doing so will break the aspect ratio of the source video, resulting in video that appears either "squashed" or "stretched" along its horizontal axis.

6.3.1.7 Video attribute

`video=track://track?hsize=number&vsize=number`

`video=http://url?hsize=number&vsize=number`

By default, the av object will display video from the current video resource playing out to the set-top-box when the browser was instantiated.

In situations where the launch point triggers a tune to a new channel or a new transponder in order to download the microbrowser code, the Browser should automatically tune back to the originating video stream after it is launched.

This means that for most services, application authors do not need to specify the video attribute of the av element; the default video stream currently being viewed will simply be played by the browser after it is instantiated.

Where necessary, the application author can individually specify the track location of a video resource to be played by the av object. Track ids are valid only within the context of a service. A track tune is achieved by specifying a service, then specifying a track within that service.

Note that tracks are defined independently of services. See the definitions for the service attribute of the av element for further information.

The location can be defined in a variety of ways:

- track:track - DVB or Network specific track specification

http://url - The URL of an MPEG video resource, resolved against the location of the current Modeset.

All of these syntaxes can also have hsize and vsize specifications appended to them. These are only required if the video resource is to be resized in the AV window.

- track://0x56494400?hsize=0.5&vsize=0.5 - This example happens to use a network specific hex track notation to define the video resource location. The syntax works in exactly the same way for number and for url notation.

The hsize and vsize values specify the video resource size relative to full-screen video, and both default to 1.0.

Note that in WML, the "&" character is a control character, and must be escaped before it can be used within the markup. The escape sequence for the "&" character is "&". The above example would therefore be written in the WML as follows:

- track://0x56494400?hsize=0.5&vsize=0.5

6.3.1.8 Audio attribute

audio=track:track

audio=http://url

This attribute defines the location of the audio resource to be delivered:

- track:track - DVB or Network specific track specification.

http://url - The URL of an MPEG video resource, resolved against the location of the current Modeset.

6.3.1.9 Toggle attribute

toggle=(key:number)

This attribute defines the key binding on the remote handset that will toggle between browser display and full screen video display.

6.3.1.10 Offset attribute

offset=x,y

If defined, the offset attribute defines in pixels the offset of the video frame from the av window defined by the rect attribute.

6.3.1.11 Bgimage attribute

bgimage="http://mysite/myimage.mpg"

The bgimage attribute defines the location of a static MPEG image to be displayed in the av object whenever MPEG streaming video is not being played.

6.3.1.12 Service attribute

service="service://network/transport/service" The service attribute defines a DVB service to tune to for display in the av object. The network, transport and service values correspond to standard DVB service specification of original network identifier, transport stream identifier and service identifier as specified in EN 300 468 [3].

It is anticipated that network specific service descriptions may also be implemented.

Note that a service consists of multiple tracks. See the video attribute of the av element for information on how to tune to a specific track within a service.

6.3.1.13 Example

```
<av mode='full' video='track:0x56494401' />
<av mode='window' video='track:0x56494400' rect='336,48,656,305' hscale='0.5' vscale='0.5'
toggle='27' />
<av mode='blank' />
```

6.3.2 Status element

diagram	 <p>Compulsory onscreen object used to display the current browser connection status</p>				
namespace	http://waptv.com/xsd/modeset				
used by	element mode				
attributes	Name	Type	Use	Default	Fixed
	id	xs:string	optional		status
	align	xs:NMTOKEN	optional	bc	
	rect	xs:string	required		
	visible	xs:NMTOKEN	optional	true	
source	<pre><xs:element name="status"> <xs:annotation> <xs:documentation>Compulsory onscreen object used to display the current browser connection status</xs:documentation> <xs:appinfo> <comp:element-def name="status" bytecode="0b"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional" fixed="status"/> <xs:attribute name="align" use="optional" default="bc"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl"/> <xs:enumeration value="tc"/> <xs:enumeration value="tr"/> <xs:enumeration value="ml"/> <xs:enumeration value="mc"/> <xs:enumeration value="mr"/> <xs:enumeration value="bl"/> <xs:enumeration value="bc"/> <xs:enumeration value="br"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="rect" type="xs:string" use="required"/> <xs:attribute name="visible" use="optional" default="true"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="true"/> <xs:enumeration value="false"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element></pre>				

The status element controls display of the online status of the browser. Each state is represented by a local source image:

- Offline and idle - offline;
- Offline and busy - offactive;
- Dialling - connect;
- Online and idle - online;
- Online and busy - onactive.

6.3.2.1 Id attribute

`id="status"`

There should be only a single status instance defined, and it must be allocated the id status.

6.3.2.2 Align attribute

`align=(tl|tc|tr|ml|mc|mr|bl|bc|br)`

The align attribute specifies the position of the status image graphic within the bounds defined for the status object. Default value is "bc".

6.3.2.3 Rect attribute

`rect=x1,y1,x2,y2`

The rect attribute defines where the status window is to be drawn on the screen.

6.3.2.4 Examples

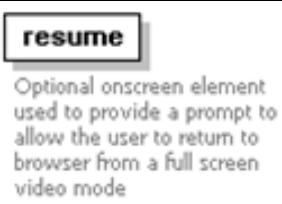
```
<status visible='false' />
```

This example disables the Status object in a layout.

```
<status id='xxxx' align='top' visible='true' rect='80,486,320,128' />
```

This example defines characteristics for the status object.

6.3.3 Resume element

diagram					
namespace	http://waptv.com/xsd/modeset				
used by	element mode				
attributes	Name	Type	Use	Default	Fixed
	id	xs:string	optional		resume
	localsrc	xs:string	optional		
	rect	xs:string	optional		
source	<pre> <xs:element name="resume"> <xs:annotation> <xs:documentation>Optional onscreen element used to provide a prompt to allow the user to return to browser from a full screen video mode</xs:documentation> <xs:appinfo> <comp:element-def name="resume" bytecode="12"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional" fixed="resume"/> <xs:attribute name="localsrc" type="xs:string" use="optional"/> <xs:attribute name="rect" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

If defined, the resume element is displayed over full screen video when the user toggles from the browser to full screen video mode. See clause 6.3.1, av element.

6.3.3.1 Id attribute

id="resume"

There can only be one resume element defined. It must have the id "resume".

6.3.3.2 Localsrc attribute

Localsrc=vdata

This attribute defines the location of the resume element resource to be displayed on screen.

6.3.3.3 Rect attribute

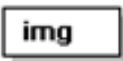
rect=x1,y1,x2,y2

The rect attribute defines where the resume window is to be drawn on the screen.

6.3.3.4 Example

```
<resume rect='500,48,656,200' localsrc='resume' />
```

6.3.4 Img element

diagram					
namespace	http://waptv.com/xsd/modeset				
used by	element mode				
attributes	Name	Type	Use	Default	Fixed
	id	xs:string	optional		
	src	xs:string	optional		
	localsrc	xs:string	optional		
	vspace	xs:string	optional	0	
	hspace	xs:string	optional	0	
	align	xs:NMTOKEN	optional	bottom	
	height	xs:string	optional		
	width	xs:string	optional		
	rect	xs:string	optional		
	visible	xs:NMTOKEN	optional	true	
	cache-control	xs:NMTOKEN	optional	cache	
source	<pre> <xs:element name="img"> <xs:annotation> <xs:appinfo> <comp:element-def name="img" bytecode="2e" /> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional" /> <xs:attribute name="src" type="xs:string" use="optional" /> <xs:attribute name="localsrc" type="xs:string" use="optional" /> <xs:attribute name="vspace" type="xs:string" use="optional" default="0" /> <xs:attribute name="hspace" type="xs:string" use="optional" default="0" /> <xs:attribute name="align" use="optional" default="bottom"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="top" /> <xs:enumeration value="middle" /> <xs:enumeration value="bottom" /> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="height" type="xs:string" use="optional" /> <xs:attribute name="width" type="xs:string" use="optional" /> <xs:attribute name="rect" type="xs:string" use="optional" /> <xs:attribute name="visible" use="optional" default="true"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="true" /> <xs:enumeration value="false" /> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="cache-control" use="optional" default="cache"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="cache" /> <xs:enumeration value="nocache" /> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>				

6.3.4.1 Id attribute

id="img"

There can be multiple img instances defined. Each should be associated with a unique id. This attribute can be manipulated from the WML content using the setattr element.

6.3.4.2 Src attribute

`src="HREF"`

The `src` attribute specifies the module in which the image resource is contained.

6.3.4.3 Localsrc attribute

`localsrc="vdata"`

The `localsrc` attribute defines the location of a local source image.

6.3.4.4 Vspace attribute

`vspace="%10"`

The `vspace` attribute specifies the vertical space to be left above and below the image. The space is expressed as a percentage of the available space.

6.3.4.5 Hspace attribute

`hspace="%10"`

The `hspace` attribute defines the horizontal space to be left on either side of the image. The space is expressed as a percentage of the available space.

6.3.4.6 Align attribute

`align=(top|middle|bottom)`

The `align` attribute specifies the alignment of the image within the flow of the WML content. The `align` attribute is specified against the current insertion point, and has three possible values:

- `bottom`: vertically align the bottom of the image with the current baseline. This is the default value.
- `middle`: vertically align the centre of the image with the centre of the current text line.
- `top`: vertically align the top of the image with the top of the current text line.

6.3.4.7 Height & width attributes

`height=length`

`width=length`

The `height` and `width` attributes specify the size of the image to the WTVML microbrowser so it can reserve space for it and continue rendering the card while waiting for the image data. This assists in optimizing a card for quick display.

6.3.4.8 Rect attribute

`rect=x1,y1,x2,y2`

The `rect` attribute defines where the image window is to be drawn on the screen using `x,y` coordinates for the top left and bottom right corners.

6.3.4.9 Visible attribute

`visible=(true|false)`

The `visible` attribute controls whether the image is displayed on screen or not. Invisible images can be used to assist the developer to layout their WML content.

6.3.4.10 Cache control attribute

cache-control="(cache|nocache)"


The cache-control attribute specifies how the browser should cache this resource. There are two possible values for the cache-control attribute. The default value is "cache".

Value	Description
cache	Keep the resource in cache until the memory is required by the browser.
nocache	Do not cache the resource.

6.3.4.11 Example

```
<img id='xxxx' src='http://193.195.2.145/ classicl.jpg' rect='550,220,600,502' />
<img id='usl' rect='180,49,280,149' />
```

6.3.5 Clock element

diagram					
namespace	http://waptv.com/xsd/modeset				
used by	element mode				
attributes	Name	Type	Use	Default	Fixed
	id	xs:string	optional	clock	
	format	xs:string	required		
	value	xs:string	optional		
	visible	xs:boolean	optional	true	
	font	xs:string	optional		
	rect	xs:string	required		
	align	xs:NMTOKEN	optional	tl	
	hpad	xs:string	optional		
	vpad	xs:string	optional		
	aacolor	xs:string	optional		
	bgcolor	xs:string	optional		
	bgradius	xs:string	optional		
	bordercolor	xs:string	optional		
	borderwidth	xs:string	optional		
source	<pre><xs:element name="clock"> <xs:annotation> <xs:documentation>Optional onscreen element to display date and time.</xs:documentation> <xs:appinfo> <comp:element-def name="clock" bytecode="14"/> </xs:appinfo> </xs:annotation> </xs:element> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional" default="clock"/> <xs:attribute name="format" type="xs:string" use="required"/> <xs:attribute name="value" type="xs:string" use="optional"/> <xs:attribute name="visible" type="xs:boolean" use="optional" default="true"/> <xs:attribute name="font" type="xs:string" use="optional"/> <xs:attribute name="rect" type="xs:string" use="required"/> <xs:attribute name="align" use="optional" default="tl"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl"/> <xs:enumeration value="tc"/> <xs:enumeration value="tr"/> <xs:enumeration value="ml"/> <xs:enumeration value="mc"/> <xs:enumeration value="mr"/> <xs:enumeration value="bl"/> <xs:enumeration value="bc"/> <xs:enumeration value="br"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType></pre>				

```

</xs:attribute>
<xs:attribute name="hpad" type="xs:string" use="optional"/>
<xs:attribute name="vpad" type="xs:string" use="optional"/>
<xs:attribute name="aacolor" type="xs:string" use="optional"/>
<xs:attribute name="bgcolor" type="xs:string" use="optional"/>
<xs:attribute name="bgradius" type="xs:string" use="optional"/>
<xs:attribute name="bordercolor" type="xs:string" use="optional"/>
<xs:attribute name="borderwidth" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>

```

The clock element can be used to place a clock on screen.

6.3.5.1 Id attribute

```
id="clock"
```

The id attribute provides the clock element with a unique name. This attribute can be manipulated from the WML content using the setattr element.

6.3.5.2 Format attribute

```
format="%d%b%H:%M:%S"
```

The default format for the clock is the same as the OpenTV system call `O_tm_to_string`. The format `%d%b%H:%M:%S` displays the time in the format '31 Mar 15:16:55'.

Format code	Description
%a	Replaced with an abbreviated weekday name appropriate to the locale (for example, in the US: Sun, ..., Sat).
%A	Replaced with full weekday name appropriate to the locale (for example, in the US: Sunday, ..., Saturday).
%b	Replaced with abbreviated month name appropriate for the locale (for example, in the US: Jan, ..., Dec).
%B	Replaced with full month name appropriate to the locale (for example, in the US: January, ..., December).
%c	Replaced by local date and time information in the format acceptable in the locale. In the US, a typical output will look like "Thu Jan 1 00:00:01 1970".
%d	Replaced by the day of the month (01 to 31).
%H	Replaced by the hour (24-hour clock) (00 to 23).
%I	Replaced by the hour (12-hour clock) (01 to 12).
%j	Replaced by the day of the year (001 to 366).
%m	Replaced by the month (01 to 12).
%M	Replaced by the minute (00 to 59).
%p	Replaced by the locale specific string for AM or PM.
%S	Replaced by the second (00 to 59).
%U	Replaced by the week number of the year (Sunday as 1st day of the week) (00 to 53).
%w	Replaced by the weekday (0 to 6, Sunday is 0).
%W	Replaced by the week number of the year (Monday as 1st day of the week) (00 to 53).
%x	Replaced by the local date representation in the current locale format. In the US, this looks like "1/1/70".
%X	Replaced by the local time representation in locale specific format. In the US, typical output looks like "13:32:21".
%y	Replaced by the year without the century. (e.g. 01).
%Y	Replaced by the year with the century. (e.g. 2001).

6.3.5.3 Value attribute

```
value=%vdata
```

The value attribute specifies the content of the text element.

6.3.5.4 Visible attribute

visible=(true|false)

The visible attribute specifies whether the element is to be visible or not.

6.3.5.5 Font attribute

font="big"

The font attribute specifies the font to be used to display the element.

6.3.5.6 Rect attribute

rect="x1,y1,x2,y2)

The rect attribute defines where the element is to be drawn on the screen.

6.3.5.7 Align attribute

align="(tl|tm|tr|cl|cm|cr|bl|bm|br)"

The align attribute specifies which corner of the object is to be anchored.


6.3.5.8 %Wcell attributes

WTVML 6.3 extensions to WML support the application of cell formatting to all elements.

6.3.5.9 Example

```
<clock id='clock' format='%d %b %H:%M:%S' font='menu' rect=' 80,486,320,128' />
```

6.3.6 Title element

diagram					
namespace	http://waptv.com/xsd/modeset				
used by	elements browser dialog				
attributes	Name	Type	Use	Default	Fixed
	id	xs:string	optional		title
	value	xs:string	optional		
	visible	xs:boolean	optional	true	
	font	xs:string	optional		
	rect	xs:string	optional		
	align	xs:NMTOKEN	optional	tl	
	hpad	xs:string	optional		
	vpad	xs:string	optional		
	aacolor	xs:string	optional		
	bgcolor	xs:string	optional		
	bgradius	xs:string	optional		
	bordercolor	xs:string	optional		
	borderwidth	xs:string	optional		
source	<pre> <xs:element name="title"> <xs:annotation> <xs:documentation>Displays content of card Title attribute</xs:documentation> <xs:appinfo> <comp:element-def name="title" bytecode="10"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional" fixed="title"/> <xs:attribute name="value" type="xs:string" use="optional"/> <xs:attribute name="visible" type="xs:boolean" use="optional" default="true"/> <xs:attribute name="font" type="xs:string" use="optional"/> <xs:attribute name="rect" type="xs:string" use="optional"/> <xs:attribute name="align" use="optional" default="tl"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl"/> <xs:enumeration value="tc"/> <xs:enumeration value="tr"/> <xs:enumeration value="ml"/> <xs:enumeration value="mc"/> <xs:enumeration value="mr"/> <xs:enumeration value="bl"/> <xs:enumeration value="bc"/> <xs:enumeration value="br"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="hpad" type="xs:string" use="optional"/> <xs:attribute name="vpad" type="xs:string" use="optional"/> <xs:attribute name="aacolor" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

The title element can be used to place a title text area on screen. If defined, the title element is used to display the value of the title attribute of the current card.

6.3.6.1 Id attribute

id="title"

There can only be one title element defined. It must have the id "title".

6.3.6.2 Value attribute

value=% vdata

The value attribute specifies the content of the text element.

6.3.6.3 Visible attribute

Visible="(true|false)"

The visible attribute specifies whether the element is to be visible or not.

6.3.6.4 Font attribute

Font=CDATA

The font attribute specifies the font to be used to display the element.

6.3.6.5 Rect attribute

rect="x1,y1,x2,y2"

The rect attribute defines where the element is to be drawn on the screen.

6.3.6.6 Align attribute

align="(tl|tm|tr|cl|cm|cr|bl|bm|br)"

The align attribute specifies which corner of the object is to be anchored.

6.3.6.7 Hpad attribute

hpad=%number

The hpad attribute defines in pixels the horizontal space to be left between the border of the table cell, and the contents of the cell.

6.3.6.8 Vpad attribute

vpad="10"

The vpad attribute defines in pixels the vertical space to be left between the border of the table cell, and the contents of the cell.

6.3.6.9 Aacolor attribute

aacolor="(#rrggbb|transp)"

The aacolor attribute defines the colour that text will be anti-aliased against.

6.3.6.10 Bgcolor attribute

bgcolor="(#rrggbb|transp)"

The bgcolor attribute defines the colour of the box that will be drawn beneath the cell element. If bgcolor is defined, text will also be anti-aliased against this colour. If it is not defined, no box will be drawn and the value of aacolor will be used for anti-aliasing.

6.3.6.11 Bgradius attribute

bgradius="10"

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the table cell.

6.3.6.12 Bordercolor attribute

bordercolor="(#rrggbb|transp)"

The bordercolor attribute defines in RGB values the colour for the cell border.

6.3.6.13 Borderwidth attribute


borderwidth="10"

The borderwidth attribute defines in pixels the width of the table cell border.

6.3.6.14 Example

```
<title rect='180,128,320,314' font='title' bgcolor='#ff0000' align='bl' />
```

6.3.7 Wtvml:precache element

diagram					
namespace	http://waptv.com/xsd/wtvml				
used by	elements mode wtvml:wml				
attributes	Name	Type	Use	Default	Fixed
	href	xs:string	required		
	cache-set	xs:string	optional		
	cache-control	xs:NMTOKEN	optional	cache	
source	<pre><xs:element name="precache"> <xs:annotation> <xs:documentation>controls cache behaviour</xs:documentation> <xs:appinfo> <comp:element-def name="precache" bytecode="105"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="href" type="xs:string" use="required"/> <xs:attribute name="cache-set" type="xs:string" use="optional"/> <xs:attribute name="cache-control" use="optional" default="cache"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="cache"/> <xs:enumeration value="no-cache"/> <xs:enumeration value="domain"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element></pre>				

The precache element can be used to precisely control browser caching behaviour. Using this element, it is possible to selectively cache and flush resources. Typically this is used to control the downloading, caching and flushing of graphic image resources to optimize user experience.

6.3.7.1 Href element

`href="http://mysite/myimage.jpg"` The href attribute specifies the URL of the resource to be precached.

Individual precache requests are added to a queue in the order of their declaration, and are fetched one at a time while the browser is idle.

6.3.7.2 Cache-set attribute

`cache-set="a"`

The cache-set attribute is used to assign a single character identifier to the cached resource. This can be any ASCII character except those used in the standard WML control character set.

The same character identifier can be assigned to multiple resources.

The default value for the cache-set attribute is 0.

It is possible to selectively flush cached resources according to their cache-set attributes by using `<setattr wtv.flushcache>`

See clause 9.3.10 for details on using the `wtv.flushcache` attribute.


6.3.8 Cache-control attribute

`cache-control="%precache-control"`

The cache-control attribute specifies how the browser should cache this resource. There are three possible values for the cache-control attribute. The default value is "cache".

Value	Description
domain	Keep the resource in cache until a domain change or until the memory is required by the browser.
cache	Keep the resource in cache until the memory is required by the browser.
no-cache	Do not cache the resource.

6.3.9 Hint element

diagram	 <p>Optional onscreen element used to provide context sensitive help</p>				
namespace	http://waptv.com/xsd/modeset				
used by	element				
attributes	Name	Type	Use	Default	Fixed
	id	xs:string	optional		hint
	value	xs:string	optional		
	visible	xs:boolean	optional	true	
	font	xs:string	optional		
	rect	xs:string	optional		
	align	xs:NMTOKEN	optional	tl	
	hpad	xs:string	optional		
	vpad	xs:string	optional		
	aacolor	xs:string	optional		
	bgcolor	xs:string	optional		
	bgradius	xs:string	optional		
	bordercolor	xs:string	optional		
	borderwidth	xs:string	optional		
source	<pre> <xs:element name="hint"> <xs:annotation> <xs:documentation>Optional onscreen element used to provide context sensitive help</xs:documentation> <xs:appinfo> <comp:element-def name="hint" bytecode="0f"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional" fixed="hint"/> <xs:attribute name="value" type="xs:string" use="optional"/> <xs:attribute name="visible" type="xs:boolean" use="optional" default="true"/> <xs:attribute name="font" type="xs:string" use="optional"/> <xs:attribute name="rect" type="xs:string" use="optional"/> <xs:attribute name="align" use="optional" default="tl"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl"/> <xs:enumeration value="tc"/> <xs:enumeration value="tr"/> <xs:enumeration value="ml"/> <xs:enumeration value="mc"/> <xs:enumeration value="mr"/> <xs:enumeration value="bl"/> <xs:enumeration value="bc"/> <xs:enumeration value="br"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="hpad" type="xs:string" use="optional"/> <xs:attribute name="vpad" type="xs:string" use="optional"/> <xs:attribute name="aacolor" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

If defined, the hint element is used to display the value of the title attribute of the currently focussed onscreen object. This is commonly used to provide context sensitive help to the user, as hint text can be associated with any focusable object. This allows a brief message to be displayed in the title area when the user has focus on an onscreen object.

6.3.9.1 Id attribute

`id="hint"`

There can only be one hint element. It must have the id "hint".

6.3.9.2 Value attribute

`value=%vdata`

The value attribute specifies the content of the text element.

6.3.9.3 Visible attribute

`Visible="(true|false)"`

The visible attribute specifies whether the element is to be visible or not.

6.3.9.4 Font attribute

`Font=CDATA`

The font attribute specifies the font to be used to display the element.

6.3.9.5 Rect attribute

`rect="x1,y1,x2,y2"`

The rect attribute defines where the element is to be drawn on the screen.

6.3.9.6 Align attribute

`align="(tl|tm|tr|cl|cm|cr|bl|bm|br)"`

The align attribute specifies which corner of the object is to be anchored.

6.3.9.7 Hpad attribute

`hpad="10"`

The hpad attribute defines in pixels the horizontal space to be left between the border of the table cell, and the contents of the cell.

6.3.9.8 Vpad attribute

`vpad="10"`

The vpad attribute defines in pixels the vertical space to be left between the border of the table cell, and the contents of the cell.

6.3.9.9 Aacolor attribute

`aacolor="(#rrggbb|transp)"`

The aacolor attribute defines the colour that text will be anti-aliased against.

6.3.9.10 Bgcolor attribute

```
bgcolor="#rrggbb|transp"
```

The bgcolor attribute defines the colour of the box that will be drawn beneath the cell element. If bgcolor is defined, text will also be anti-aliased against this colour. If it is not defined, no box will be drawn and the value of aacolor will be used for anti-aliasing.

6.3.9.11 Bgradius attribute

```
bgradius="10"
```

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the table cell.

6.3.9.12 Bordercolor attribute

```
bordercolor="#rrggbb|transp)"
```

The bordercolor attribute defines in RGB values the colour for the cell border.

6.3.9.13 Borderwidth attribute


```
borderwidth="10"
```

The borderwidth attribute defines in pixels the width of the table cell border.

6.3.9.14 Example

```
<hint id='hint' rect='150,470,650,550' align='tl' font='hint' bgcolor='' visible="true"/>
```

6.3.10 Callrate element

diagram	 <p>Compulsory onscreen object used to display the phone tariff rate for an online connection</p>																																																																						
namespace	http://waptv.com/xsd/modeset																																																																						
used by	element mode																																																																						
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>xs:string</td> <td>optional</td> <td></td> <td>callrate</td> </tr> <tr> <td>value</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>visible</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>true</td> <td></td> </tr> <tr> <td>font</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>rect</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>align</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>tl</td> <td></td> </tr> <tr> <td>hpad</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>vpad</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>aacolor</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bgcolor</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bgradius</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bordercolor</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>borderwidth</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	id	xs:string	optional		callrate	value	xs:string	optional			visible	xs:NMTOKEN	optional	true		font	xs:string	optional			rect	xs:string	required			align	xs:NMTOKEN	optional	tl		hpad	xs:string	optional			vpad	xs:string	optional			aacolor	xs:string	optional			bgcolor	xs:string	optional			bgradius	xs:string	optional			bordercolor	xs:string	optional			borderwidth	xs:string	optional		
Name	Type	Use	Default	Fixed																																																																			
id	xs:string	optional		callrate																																																																			
value	xs:string	optional																																																																					
visible	xs:NMTOKEN	optional	true																																																																				
font	xs:string	optional																																																																					
rect	xs:string	required																																																																					
align	xs:NMTOKEN	optional	tl																																																																				
hpad	xs:string	optional																																																																					
vpad	xs:string	optional																																																																					
aacolor	xs:string	optional																																																																					
bgcolor	xs:string	optional																																																																					
bgradius	xs:string	optional																																																																					
bordercolor	xs:string	optional																																																																					
borderwidth	xs:string	optional																																																																					
source	<pre> <xs:element name="callrate"> <xs:annotation> <xs:documentation>Compulsory onscreen object used to display the phone tarrif rate for an online connection</xs:documentation> </xs:annotation> <xs:appinfo> <comp:element-def name="callrate" bytecode="11"/> </xs:appinfo> </xs:element> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional" fixed="callrate"/> <xs:attribute name="value" type="xs:string" use="optional"/> <xs:attribute name="visible" use="optional" default="true"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="true"/> <xs:enumeration value="false"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="font" type="xs:string" use="optional"/> <xs:attribute name="rect" type="xs:string" use="required"/> <xs:attribute name="align" use="optional" default="tl"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl"/> <xs:enumeration value="tc"/> <xs:enumeration value="tr"/> <xs:enumeration value="ml"/> <xs:enumeration value="mc"/> <xs:enumeration value="mr"/> <xs:enumeration value="bl"/> <xs:enumeration value="bc"/> <xs:enumeration value="br"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="hpad" type="xs:string" use="optional"/> <xs:attribute name="vpad" type="xs:string" use="optional"/> <xs:attribute name="aacolor" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>																																																																						

If defined, the callrate element is used to display the callrate text for the current domain when the browser goes online.

6.3.10.1 Id attribute

`id="callrate"`

There can only be one callrate element defined. It must have the id "callrate".

6.3.10.2 Value attribute

`value=%vdata`

The value attribute specifies the content of the text element.

6.3.10.3 Visible attribute

`Visible="(true|false)"`

The visible attribute specifies whether the element is to be visible or not.

6.3.10.4 Font attribute

`Font=CDATA`

The font attribute specifies the font to be used to display the element.

6.3.10.5 Rect attribute

`rect="x1,y1,x2,y2"`

The rect attribute defines where the element is to be drawn on the screen.

6.3.10.6 Align attribute

`align="(tl|tm|tr|cl|cm|cr|bl|bm|br)"`

The align attribute specifies which corner of the object is to be anchored.

6.3.10.7 Hpad attribute

`hpad="10"`

The hpad attribute defines in pixels the horizontal space to be left between the border of the table cell, and the contents of the cell.

6.3.10.8 Vpad attribute

`vpad="10"`

The vpad attribute defines in pixels the vertical space to be left between the border of the table cell, and the contents of the cell.

6.3.10.9 Aacolor attribute

`aacolor="(#rrggbb|transp)"`

The aacolor attribute defines the colour that text will be anti-aliased against.

6.3.10.10 Bgcolor attribute

```
bgcolor="#rrggbb|transp"
```

The bgcolor attribute defines the colour of the box that will be drawn beneath the cell element. If bgcolor is defined, text will also be anti-aliased against this colour. If it is not defined, no box will be drawn and the value of aacolor will be used for anti-aliasing.

6.3.10.11 Bgradius attribute

```
bgradius="10"
```

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the table cell.

6.3.10.12 Bordercolor attribute

```
bordercolor="#rrggbb|transp)"
```

The bordercolor attribute defines in RGB values the colour for the cell border.

6.3.10.13 Borderwidth attribute


```
borderwidth="10"
```

The borderwidth attribute defines in pixels the width of the table cell border.

6.3.10.14 Example

```
<callrate rect='180,78,320,128' align='bl' font='callrate' bgcolor='#07084f' />
```

6.3.11 Text element

diagram					
namespace	http://waptv.com/xsd/modeset				
used by	element mode				
attributes	Name	Type	Use	Default	Fixed
	id	xs:string	optional		
	value	xs:string	optional		
	visible	xs:boolean	optional	true	
	font	xs:string	optional		
	rect	xs:string	optional		
	align	xs:NMTOKEN	optional	tl	
	hpad	xs:string	optional		
	vpad	xs:string	optional		
	aacolor	xs:string	optional		
	bgcolor	xs:string	optional		
	bgradius	xs:string	optional		
	bordercolor	xs:string	optional		
	borderwidth	xs:string	optional		
	borderfocused	xs:string	optional		
source	<pre> <xs:element name="text"> <xs:annotation> <xs:documentation>Defines default text characteristics</xs:documentation> <xs:appinfo> <comp:element-def name="text" bytecode="0c"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional"/> <xs:attribute name="value" type="xs:string" use="optional"/> <xs:attribute name="visible" type="xs:boolean" use="optional" default="true"/> <xs:attribute name="font" type="xs:string" use="optional"/> <xs:attribute name="rect" type="xs:string" use="optional"/> <xs:attribute name="align" use="optional" default="tl"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl"/> <xs:enumeration value="tc"/> <xs:enumeration value="tr"/> <xs:enumeration value="ml"/> <xs:enumeration value="mc"/> <xs:enumeration value="mr"/> <xs:enumeration value="bl"/> <xs:enumeration value="bc"/> <xs:enumeration value="br"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="hpad" type="xs:string" use="optional"/> <xs:attribute name="vpad" type="xs:string" use="optional"/> <xs:attribute name="aacolor" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> <xs:attribute name="borderfocused" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

The text element is used to display formatted text onscreen.

6.3.11.1 Id attribute

`id="text"`

The id attribute provides the element with a unique name. This attribute can be manipulated from the WML content using the `setattr` element.

6.3.11.2 Value attribute

`value=%vdata`

The value attribute specifies the content of the text element.

Certain special characters must be "escaped" in order to appear correctly onscreen in a text object.

Character	Escape code
>	>.
\$	&\$.
'	&apos.
"	".

6.3.11.3 Visible attribute

`Visible="(true|false)"`

The visible attribute specifies whether the element is to be visible or not.

6.3.11.4 Font attribute

`Font=CDATA`

The font attribute specifies the font to be used to display the element.

6.3.11.5 Rect attribute

`rect="x1,y1,x2,y2"`

The rect attribute defines where the element is to be drawn on the screen.

6.3.11.6 Align attribute

`align="(tl|tm|tr|cl|cm|cr|bl|bm|br)"`

The align attribute specifies which corner of the object is to be anchored.

6.3.11.7 Hpad attribute

`hpad="10"`

The hpad attribute defines in pixels the horizontal space to be left between the border of the table cell, and the contents of the cell.

6.3.11.8 Vpad attribute

`vpad="10"`

The vpad attribute defines in pixels the vertical space to be left between the border of the table cell, and the contents of the cell.

6.3.11.9 Aacolor attribute

```
aacolor="(#rrggbb|transp)"
```

The aacolor attribute defines the colour that text will be anti-aliased against.

6.3.11.10 Bgcolor attribute

```
bgcolor="(#rrggbb|transp)"
```

The bgcolor attribute defines the colour of the box that will be drawn beneath the cell element. If bgcolor is defined, text will also be anti-aliased against this colour. If it is not defined, no box will be drawn and the value of aacolor will be used for anti-aliasing.

6.3.11.11 Bgradius attribute

```
bgradius="10"
```

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the table cell.

6.3.11.12 Bordercolor attribute

```
bordercolor="(#rrggbb|transp)"
```

The bordercolor attribute defines in RGB values the colour for the cell border.

6.3.11.13 Borderwidth attribute

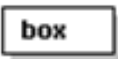
```
borderwidth="10"
```

The borderwidth attribute defines in pixels the width of the table cell border.

6.3.11.14 Example

```
<text id='dynamic' rect=' 180,504,660,532' />
```

6.3.12 Box element

diagram					
namespace	http://waptv.com/xsd/modeset				
attributes	Name	Type	Use	Default	Fixed
	id	xs:string	optional	box	
	visible	xs:boolean	optional	true	
	rect	xs:string	required		
	bgcolor	xs:string	optional		
	bgradius	xs:string	optional		
	borderwidth	xs:string	optional		
	bordercolor	xs:string	optional		
source	<pre> <xs:element name="box"> <xs:annotation> <xs:appinfo> <comp:element-def name="box" bytecode="106"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional" default="box"/> <xs:attribute name="visible" type="xs:boolean" use="optional" default="true"/> <xs:attribute name="rect" type="xs:string" use="required"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

The box element can be used to draw a primitive box graphic on the OSD layer of the set-top-box display.

6.3.12.1 Id attribute

id="vdata"

The id element is used to define a unique ID for the box object. Box attribute values can then be manipulated using setattr.

6.3.12.2 Visible attribute

visible="true|false"

The visible attribute specifies whether or not the box object is visible onscreen.

6.3.12.3 Rect attribute

rect="x1,y1,x2,y2"

The rect attribute specifies the size and position of the box by specifying the x,y coordinates of the top left and bottom right corners of the box.

6.3.12.4 Bgcolor attribute

bgcolor="#rrggbb"

The bgcolor attribute defines the background colour for the box in RGB values.

6.3.12.5 Bgradius attribute

bgradius="25"

The bgradius attribute defines in pixels the radius to which the corners of the box will be rounded when drawn onscreen.

6.3.12.6 Borderwidth attribute

borderwidth="5"

The borderwidth attribute defines in pixels the width of the box object border.

6.3.12.7 Bordercolor attribute

bordercolor="#rrggbb"

The bordercolor attribute defined in RGB values the colour for the box object border.

6.3.13 Do elements

Two different kinds of do element are supported in modeset XML files. They are distinguished by their respective namespaces: modeset:do and wtvml:do.

The native modeset:do is used to specify default appearance characteristics of do elements within the menu object. The modeset:do can only be implemented by nesting inside a parent menu element in a modeset.

Although the wtvml:do is native to wtvml, and is defined within the wtvml.xsd definition, it can also be implemented within modeset XML files. When used in this way, the wtvml:do specifies an actionable link that will appear in the menu for all cards under this mode, unless it is explicitly overwritten by a menu do of the same type at card level.

6.3.13.1 Wtvml:do element

diagram					
namespace	http://waptv.com/xsd/wtvml				
children	wtvml:go wtvml:prev wtvml:noop wtvml:refresh				
used by	elements mode wtvml:b wtvml:big wtvml:card wtvml:em wtvml:fieldset wtvml:font wtvml:i wtvml:p wtvml:pre wtvml:small wtvml:strong wtvml:td wtvml:template wtvml:u				
attributes	Name	Type	Use	Default	Fixed
	type	xs:NMTOKEN	required		
	label	xs:string	optional		
	name	xs:NMTOKEN	optional		
	optional	xs:boolean	optional	false	
	tabindex	xs:NMTOKEN	optional		
	title	xs:string	optional		
	focus	xs:boolean	optional	true	
	id	xs:string	optional		
	class	xs:string	optional		
	icon	xs:string	optional		
	localsrc	xs:string	optional		
	font	xs:string	optional		
	bglocalsrc	xs:string	optional		
	bgoffset	xs:string	optional		
	pos	xs:string	optional		
source	<pre><xs:element name="do"> <xs:annotation> <xs:appinfo> <comp:element-def name="do" bytecode="28"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0"> <xs:element ref="wtvml:go" minOccurs="0"/> <xs:element ref="wtvml:prev" minOccurs="0"/> <xs:element ref="wtvml:noop" minOccurs="0"/> </xs:choice> </xs:complexType> </xs:element></pre>				

```

<xs:element ref="wtvml:refresh" minOccurs="0"/>
</xs:choice>
<xs:attribute name="type" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="accept"/>
      <xs:enumeration value="prev"/>
      <xs:enumeration value="help"/>
      <xs:enumeration value="reset"/>
      <xs:enumeration value="options"/>
      <xs:enumeration value="delete"/>
      <xs:enumeration value="unknown"/>
      <xs:enumeration value="vnd.wtv-red"/>
      <xs:enumeration value="vnd.wtv-r"/>
      <xs:enumeration value="vnd.wtv-green"/>
      <xs:enumeration value="vnd.wtv-g"/>
      <xs:enumeration value="vnd.wtv-yellow"/>
      <xs:enumeration value="vnd.wtv-y"/>
      <xs:enumeration value="vnd.wtv-blue"/>
      <xs:enumeration value="vnd.wtv-b"/>
      <xs:enumeration value="vnd.wtv-sel"/>
      <xs:enumeration value="vnd.wtv-p"/>
      <xs:enumeration value="vnd.wtv-m"/>
      <xs:enumeration value="vnd.wtv-text"/>
      <xs:enumeration value="vnd.wtv-i"/>
      <xs:enumeration value="vnd.wtv-0"/>
      <xs:enumeration value="vnd.wtv-1"/>
      <xs:enumeration value="vnd.wtv-2"/>
      <xs:enumeration value="vnd.wtv-3"/>
      <xs:enumeration value="vnd.wtv-4"/>
      <xs:enumeration value="vnd.wtv-5"/>
      <xs:enumeration value="vnd.wtv-6"/>
      <xs:enumeration value="vnd.wtv-7"/>
      <xs:enumeration value="vnd.wtv-8"/>
      <xs:enumeration value="vnd.wtv-9"/>
      <xs:enumeration value="vnd.wtv-up"/>
      <xs:enumeration value="vnd.wtv-down"/>
      <xs:enumeration value="vnd.wtv-left"/>
      <xs:enumeration value="vnd.wtv-right"/>
      <xs:enumeration value="vnd.wtv-home"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="label" type="xs:string" use="optional"/>
<xs:attribute name="name" type="xs:NMTOKEN" use="optional"/>
<xs:attribute name="optional" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="tabindex" type="xs:NMTOKEN" use="optional"/>
<xs:attribute name="title" type="xs:string" use="optional"/>
<xs:attribute name="focus" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="id" type="xs:string" use="optional"/>
<xs:attribute name="class" type="xs:string" use="optional"/>
<xs:attribute name="icon" type="xs:string" use="optional"/>
<xs:attribute name="localsrc" type="xs:string" use="optional"/>
<xs:attribute name="font" type="xs:string" use="optional"/>
<xs:attribute name="bglocalsrc" type="xs:string" use="optional"/>
<xs:attribute name="bgoffset" type="xs:string" use="optional"/>
<xs:attribute name="pos" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>

```

wtvml:dos are found "loose" within a modeset XML file. These are actionable do elements and are used to populate the menu pane with actionable links.

Menu dos are sorted alphabetically by their name attribute. This allows application authors to customize the order of menu dos independently of the table and tabindex attributes.

The wtvml menu do element is an extended version of the standard WAP Forum WML do. Only additional attributes are defined here. Application authors are advised to refer to the WAP Forum WML 1.3 specification for details on standard do attributes.

6.3.13.1.1 Title attribute

title="CDATA"

The title attribute is used to specify text content to be displayed in the title pane when this do has focus.

This provides a useful mechanism to deliver context-sensitive instructions and prompts to the user. For example, a do could be labelled "Send Message" and have a title of "Press Select to send your message now". The title text will only be visible when this do has focus, thus providing additional help to the user.

6.3.13.1.2 Focus attribute

focus="(true|false)"

The focus attribute is used to control whether or not the user can navigate focus to this do element. This can be useful to permit certain do actions to be activated only by explicit key binding actions.

In the following example, the index card can only be accessed through pressing the red button on the remote control.

6.3.13.1.3 Localsrc attribute

localsrc=CDATA

The localsrc attribute has been deprecated from wtvml:do in version 6 of the microbrowser.

The microBrowser should honour the localsrc attribute to ensure backward compatibility, but future development must use the new icon attribute in place of localsrc.

6.3.13.1.4 Icon attribute

The microBrowser should attach a localsource image to most actionable dos. The image corresponds to the dotype specified for the do. A do with a dotype of vnd.wtv-g will have the "green" button icon appear onscreen beside the do.

By default, dos of "unknown" dotype will have the "a" icon image drawn on screen beside the do.

Application authors can replace the default image with another localsrc image, including images of their own design, or with a blank image, using the icon attribute.

To remove the default icon from the do, specify icon="blank", or icon="" in the do element.

6.3.13.1.5 Bglocalsrc attribute

bglocalsrc=CDATA

The bglocalsrc attribute can be used to define a local source image to appear behind each do in the menu.

6.3.13.1.6 Bgoffset attribute

bgoffset="x,y"

The bgoffset attribute is used to define the x and y offset, relative to the do's top left corner where the underlying image should be placed.

6.3.13.1.7 Pos Attribute


pos="x1,y1,x2,y2"

The pos attribute, where specified, sets the x,y coordinates for a "floating" do. This attribute only applies to wtvml:do elements used within the context of a modeset.

6.3.13.1.8 Example

```
<do id='v3' type='vnd.wtv-r' label='Home' focus='false'>
  <go href='http://xxxx.waptv.co.uk/index.wml' />
</do>
```

6.3.13.2 Do element

diagram	 <p>Defines appearance of the actionable do items that will appear in the menu</p>				
namespace	http://waptv.com/xsd/modeset				
used by	element menu				
attributes	Name	Type	Use	Default	Fixed
	font	xs:string	required		
	bgcolor	xs:string	optional		
	bglocalsrc	xs:string	optional		
	bgoffset	xs:string	optional		
	align	xs:NMTOKEN	optional	tl	
	localsrc	xs:string	optional		
	pos	xs:string	optional		
source	<pre><xs:element name="do"> <xs:annotation> <xs:appinfo> <comp:element-def name="do" bytecode="28"/> </xs:appinfo> <xs:documentation>Defines appearance of the actionable do items that will appear in the menu</xs:documentation> </xs:annotation> <xs:complexType> <xs:attribute name="font" type="xs:string" use="required"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bglocalsrc" type="xs:string" use="optional"/> <xs:attribute name="bgoffset" type="xs:string" use="optional"/> <xs:attribute name="align" use="optional" default="tl"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl"/> <xs:enumeration value="tc"/> <xs:enumeration value="tr"/> <xs:enumeration value="ml"/> <xs:enumeration value="mc"/> <xs:enumeration value="mr"/> <xs:enumeration value="bl"/> <xs:enumeration value="bc"/> <xs:enumeration value="br"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="localsrc" type="xs:string" use="optional"/> <xs:attribute name="pos" type="xs:string" use="optional"/> </xs:complexType> </xs:element></pre>				

A WTVML microbrowser may accommodate a permanently visible Options Menu. The menu comprises a structured list of do elements.

The appearance of this menu list is defined by the native modeset:do element.

6.3.13.2.1 Font attribute

Font=CDATA

The font attribute specifies the font to be used to display the text for the do element.

6.3.13.2.2 Bgcolor attribute

`bgcolor="#rrggbb|transp"`

The bgcolor attribute defines the colour of the box that will be drawn beneath the do element. If bgcolor is defined, text will also be anti-aliased against this colour. If it is not defined, no box will be drawn and the value of the aacolor attribute from the menu will be used for anti-aliasing.

6.3.13.2.3 Bglocalsrc attribute

`bglocalsrc=CDATA`

The bglocalsrc attribute can be used to define a local source image to appear behind each do in the menu.

6.3.13.2.4 Bgoffset attribute

`bgoffset="x,y"`

The bgoffset attribute is used to define the x and y offset, relative to the do's top left corner where the underlying image should be placed.

6.3.13.2.5 Align element

`align="tl"`

The align element defines which corner of the do is anchored.

6.3.13.2.6 Localsrc attribute

`localsrc=href`

The localsrc attribute can be used to override the default image associated with a do type (e.g. green button image with `doctype=vnd.wtv-g`).

6.3.13.2.7 Example

```
<menu visible='true' rect='218,464,506,498' font='menu' align='bl' rowgap='10' >
  <do localsrc='menured' pos='500,0' font='menu' bgcolor='' />
</menu>
```

6.3.14 Table element

diagram	<p>Defines default characteristics of Table</p> <p>Defines appearance Table Row attributes. Can be used within Menu object.</p>					
namespace	http://waptv.com/xsd/modeset					
children	modeset:tr					
used by	elements browser dialog					
attributes	Name	Type	Use	Default	Fixed	
	align	xs:string	optional			
	columns	xs:string	optional			
	rowgap	xs:string	optional			
	colgap	xs:string	optional			
	vspace	xs:string	optional			
	hspace	xs:string	optional			
	hpad	xs:string	optional			
	vpad	xs:string	optional			
	colwidths	xs:string	optional			
	bgimage	xs:string	optional			
	bgcolor	xs:string	optional			
	bgradius	xs:string	optional			
	bordercolor	xs:string	optional			
	borderwidth	xs:string	optional			
source	<pre> <xs:element name="table"> <xs:annotation> <xs:documentation>Defines default characteristics of Table</xs:documentation> </xs:annotation> <xs:appinfo> <comp:element-def name="table" bytecode="1f"/> </xs:appinfo> </xs:element> <xs:complexType> <xs:sequence> <xs:element ref="modeset:tr" minOccurs="0"/> </xs:sequence> <xs:attribute name="align" type="xs:string" use="optional"/> <xs:attribute name="columns" type="xs:string" use="optional"/> <xs:attribute name="rowgap" type="xs:string" use="optional"/> <xs:attribute name="colgap" type="xs:string" use="optional"/> <xs:attribute name="vspace" type="xs:string" use="optional"/> <xs:attribute name="hspace" type="xs:string" use="optional"/> <xs:attribute name="hpad" type="xs:string" use="optional"/> <xs:attribute name="vpad" type="xs:string" use="optional"/> <xs:attribute name="colwidths" type="xs:string" use="optional"/> <xs:attribute name="bgimage" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>					

A single table element may occur as a subelement of a browser. The table subelement may contain a single tr subelement which can contain a single td subelement.

In this context, the table, tr and td elements are used to define default table, row and cell characteristics that will be inherited by all WTVML tables displayed in this mode.

Default table, row and cell definitions are shared by all browsers defined in the mode.

Individual table row and cell attributes can be overwritten in the wtvml at the card level if required.

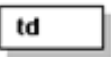
6.3.15 Tr element

diagram	 <p>Defines appearance Table Row attributes. Can be used within Menu object.</p> <p>Defines appearance of Table Cell attributes. Can be used within Menu object.</p>				
namespace	http://waptv.com/xsd/modeset				
children	modeset:td				
used by	elements menu table				
attributes	Name	Type	Use	Default	Fixed
	title	xs:string	optional		
	height	xs:string	optional		
	localsrc	xs:string	optional		
	bgimage	xs:string	optional		
	bgradius	xs:string	optional		
	bgcolor	xs:string	optional		
	bgfocused	xs:string	optional		
	bgactivated	xs:string	optional		
	borderwidth	xs:string	optional		
	bordercolor	xs:string	optional		
	borderfocused	xs:string	optional		
	borderactivated	xs:string	optional		
	hspace	xs:string	optional		
	vspace	xs:string	optional		
source	<pre> <xs:element name="tr"> <xs:annotation> <xs:documentation>Defines appearance Table Row attributes. Can be used within Menu object.</xs:documentation> </xs:annotation> <xs:appinfo> <comp:element-def name="tr" bytecode="1e"/> </xs:appinfo> </xs:element> <xs:complexType> <xs:sequence> <xs:element ref="modeset:td" minOccurs="0"/> </xs:sequence> <xs:attribute name="title" type="xs:string" use="optional"/> <xs:attribute name="height" type="xs:string" use="optional"/> <xs:attribute name="localsrc" type="xs:string" use="optional"/> <xs:attribute name="bgimage" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bgfocused" type="xs:string" use="optional"/> <xs:attribute name="bgactivated" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderfocused" type="xs:string" use="optional"/> <xs:attribute name="borderactivated" type="xs:string" use="optional"/> <xs:attribute name="hspace" type="xs:string" use="optional"/> <xs:attribute name="vspace" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

A single tr element can be used as a subelement of table within a browser element in order to set table defaults for WTVML tables displayed in this mode.

A single tr element can also be used as a subelement of menu, where it defines default appearance and behaviour of menu items within the menu object. The menu object can be considered to be a special instance of a table.

6.3.16 Td element

Diagram	 <p>Defines appearance of Table Cell attributes. Can be used within Menu object.</p>				
Namespace used by	http://waptv.com/xsd/modeset				
Attributes	Name	Type	Use	Default	Fixed
	title	xs:string	optional		
	height	xs:string	optional		
	width	xs:string	optional		
	hspace	xs:string	optional		
	vspace	xs:string	optional		
	hpad	xs:string	optional		
	vpad	xs:string	optional		
	linegap	xs:string	optional		
	colspan	xs:string	optional	1	
	rowspan	xs:string	optional	1	
	localsrc	xs:string	optional		
	bgimage	xs:string	optional		
	bgradius	xs:string	optional		
	bgcolor	xs:string	optional		
	bgfocused	xs:string	optional		
	bgactivated	xs:string	optional		
	borderwidth	xs:string	optional		
	bordercolor	xs:string	optional		
	borderfocused	xs:string	optional		
	borderactivated	xs:string	optional		
	mode	xs:NMTOKEN	optional	wrap	
source	<pre> <xs:element name="td"> <xs:annotation> <xs:documentation>Defines appearance of Table Cell attributes. Can be used within Menu object.</xs:documentation> <xs:appinfo> <comp:element-def name="td" bytecode="ld"/> <comp:allowed-element name="#PCDATA"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="title" type="xs:string" use="optional"/> <xs:attribute name="height" type="xs:string" use="optional"/> <xs:attribute name="width" type="xs:string" use="optional"/> <xs:attribute name="hspace" type="xs:string" use="optional"/> <xs:attribute name="vspace" type="xs:string" use="optional"/> <xs:attribute name="hpad" type="xs:string" use="optional"/> <xs:attribute name="vpad" type="xs:string" use="optional"/> <xs:attribute name="linegap" type="xs:string" use="optional"/> <xs:attribute name="colspan" type="xs:string" use="optional" default="1"/> <xs:attribute name="rowspan" type="xs:string" use="optional" default="1"/> <xs:attribute name="localsrc" type="xs:string" use="optional"/> <xs:attribute name="bgimage" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bgfocused" type="xs:string" use="optional"/> <xs:attribute name="bgactivated" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderfocused" type="xs:string" use="optional"/> <xs:attribute name="borderactivated" type="xs:string" use="optional"/> <xs:attribute name="mode" use="optional" default="wrap"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="wrap"/> <xs:enumeration value="nowrap"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>				

A single td element can be used as a subelement within a tr element in mode level menu and table definitions.

Depending on context, the td element can be used within a mode to define default appearance and behaviour of menu cells or WTVML table cells.

6.3.17 Menu element

Diagram																																																																																	
namespace	http://waptv.com/xsd/modeset																																																																																
children	modeset:do modeset:tr modeset:popup																																																																																
used by	element mode																																																																																
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>xs:string</td> <td>optional</td> <td>menu</td> <td></td> </tr> <tr> <td>visible</td> <td>xs:boolean</td> <td>optional</td> <td>true</td> <td></td> </tr> <tr> <td>rect</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>align</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>tl</td> <td></td> </tr> <tr> <td>rowgap</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>font</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>aacolor</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bgcolor</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bgradius</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>borderwidth</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bordercolor</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>hpad</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>vpad</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>hspace</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>vspace</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	id	xs:string	optional	menu		visible	xs:boolean	optional	true		rect	xs:string	required			align	xs:NMTOKEN	optional	tl		rowgap	xs:string	optional			font	xs:string	optional			aacolor	xs:string	optional			bgcolor	xs:string	optional			bgradius	xs:string	optional			borderwidth	xs:string	optional			bordercolor	xs:string	optional			hpad	xs:string	optional			vpad	xs:string	optional			hspace	xs:string	optional			vspace	xs:string	optional		
Name	Type	Use	Default	Fixed																																																																													
id	xs:string	optional	menu																																																																														
visible	xs:boolean	optional	true																																																																														
rect	xs:string	required																																																																															
align	xs:NMTOKEN	optional	tl																																																																														
rowgap	xs:string	optional																																																																															
font	xs:string	optional																																																																															
aacolor	xs:string	optional																																																																															
bgcolor	xs:string	optional																																																																															
bgradius	xs:string	optional																																																																															
borderwidth	xs:string	optional																																																																															
bordercolor	xs:string	optional																																																																															
hpad	xs:string	optional																																																																															
vpad	xs:string	optional																																																																															
hspace	xs:string	optional																																																																															
vspace	xs:string	optional																																																																															
source	<pre> <xs:element name="menu"> <xs:annotation> <xs:appinfo> <comp:element-def name="menu" bytecode="0a" /> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:all> <xs:element ref="modeset:do" minOccurs="0" /> <xs:element ref="modeset:tr" minOccurs="0" /> <xs:element ref="modeset:popup" minOccurs="0" maxOccurs="unbounded" /> </xs:all> <xs:attribute name="id" type="xs:string" use="optional" default="menu" /> <xs:attribute name="visible" type="xs:boolean" use="optional" default="true" /> <xs:attribute name="rect" type="xs:string" use="required" /> <xs:attribute name="align" use="optional" default="tl"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl" /> <xs:enumeration value="tc" /> <xs:enumeration value="tr" /> <xs:enumeration value="ml" /> <xs:enumeration value="mc" /> <xs:enumeration value="mr" /> <xs:enumeration value="bl" /> <xs:enumeration value="bc" /> <xs:enumeration value="br" /> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="rowgap" type="xs:string" use="optional" /> <xs:attribute name="font" type="xs:string" use="optional" /> <xs:attribute name="aacolor" type="xs:string" use="optional" /> <xs:attribute name="bgcolor" type="xs:string" use="optional" /> <xs:attribute name="bgradius" type="xs:string" use="optional" /> <xs:attribute name="borderwidth" type="xs:string" use="optional" /> <xs:attribute name="bordercolor" type="xs:string" use="optional" /> </pre>																																																																																

<pre> <xs:attribute name="hpad" type="xs:string" use="optional"/> <xs:attribute name="vpad" type="xs:string" use="optional"/> <xs:attribute name="hspace" type="xs:string" use="optional"/> <xs:attribute name="vspace" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>
--

The menu element defines the layout of the Options Menu, and the dos within it.

The Options Menu is used to provide both launch point- and application-specific navigation services. The look and feel of the Options Menu can be customized to suit the application.

The menu can contain a single tr subelement, which in turn can contain a single td element. The attributes for these subelements define the default attributes for each row and cell in the menu object.

The menu can optionally contain one or more popupdo child elements. The popupdo element is used to aggregate one or more dos into a popup menu.

6.3.17.1 Id attribute

id="menu"

There should only be a single instance of the menu defined, and it will be allocated the id menu.

6.3.17.2 Visible element

visible="true"

Determines whether the Menu is visible onscreen or not. Typically an Options menu is permanently available.

6.3.17.3 Rect attribute

rect=x1,y1,x2,y2

The rect attribute defines where the menu pane is to be drawn on the screen.

6.3.17.4 Align element

align="tl"

The align element defines which corner of the menu rectangle is anchored.

6.3.17.5 Rowgap attribute

rowgap=integer

The rowgap attribute defines the space between each do in the menu.

6.3.17.6 Font attribute

Font=CDATA

The font attribute specifies the default font to be used to display the text for the do elements in the menu. The specified font will be used unless the individual do specifies a different font.

6.3.17.7 Aacolor attribute

aacolor="(#rrggbb|transp)"

The aacolor attribute defines the colour that all the text elements in the menu will be anti-aliased against.

6.3.17.8 Bgcolor attribute

```
bgcolor="(#rrggbb|transp)"
```

The bgcolor attribute defines the background colour of the menu. If bgcolor is defined, the text of the do elements in the menu will also be anti-aliased against this colour. If it is not defined, no box will be drawn and the value of aacolor will be used for anti-aliasing.

6.3.17.9 Bgradius attribute

```
bgradius="10"
```

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the menu object.

6.3.17.10 Bordercolor attribute

```
bordercolor="(#rrggbb|transp)"
```

The bordercolor attribute defines in RGB values the colour for the cell border.

6.3.17.11 Borderwidth attribute

```
borderwidth="10"
```

The borderwidth attribute defines in pixels the width of the table cell border.

6.3.17.12 Hpad attribute

```
hpad="10"
```

The hpad attribute defines in pixels the horizontal space to be left between the border of the menu object, and the contents of the menu.

6.3.17.13 Vpad attribute

```
vpad="10"
```

The vpad attribute defines in pixels the vertical space to be left between the border of the menu object, and the contents of the menu.

6.3.17.14 Mode Attribute

```
mode=(wrap|nowrap)
```

The mode attribute defines default behaviour for text content within a table cell.

Default value is "wrap".

6.3.17.15 Example 1

```
<menu visible='true' rect='218,464,506,498' font='menu' align='bl' rowgap='10' hpad='10' vpad='10'
bgradius='20' borderwidth='2'
bordercolor='#EBEBEB' />
```

6.3.17.16 Example 2

```
<menu font='menu' rect='73,107,242,435' align='tl' bgcolor='#0a4494' rowgap='8' visible="true">
  <do localsrc='menubox2' pos='10,0' font='menu' />
  <tr>
    <td bgcolor='#ff0000' bgfocused='#0000ff' bgactivated='#00ff00'>
    </td>
  </tr>
</menu>
```

6.3.18 Popupdo element

diagram	<p>The diagram illustrates the 'popupdo' element structure. It is a rectangular box containing a smaller box with a grid pattern, representing a complex type. This complex type contains two elements: 'modeset:do' (a dashed box) and 'modeset:popupmenu' (a dashed box with a plus sign in the corner). The 'modeset:do' element is positioned above 'modeset:popupmenu'.</p>					
namespace	http://waptv.com/xsd/modeset					
children	modeset:do modeset:popupmenu					
used by	element menu					
attributes	Name	Type	Use	Default	Fixed	
	id	xs:ID	optional			
	type	xs:NMTOKEN	required			
	label	xs:string	optional			
	title	xs:string	optional			
	font	xs:string	optional			
	aacolor	xs:string	optional			
	pos	xs:string	optional			
	bglocalsrc	xs:string	optional			
	icon	xs:string	optional			
	bgoffset	xs:string	optional			
source	<pre> <xs:element name="popupdo"> <xs:annotation> <xs:appinfo> <comp:element-def name="popupdo" bytecode="108" /> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:all minOccurs="0"> <xs:element ref="modeset:do" minOccurs="0" /> <xs:element ref="modeset:popupmenu" minOccurs="0" /> </xs:all> <xs:attribute name="id" type="xs:ID" use="optional" /> <xs:attribute name="type" use="required"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="accept" /> <xs:enumeration value="prev" /> <xs:enumeration value="help" /> <xs:enumeration value="reset" /> <xs:enumeration value="options" /> <xs:enumeration value="delete" /> <xs:enumeration value="unknown" /> <xs:enumeration value="vnd.wtv-r" /> <xs:enumeration value="vnd.wtv-red" /> <xs:enumeration value="vnd.wtv-g" /> <xs:enumeration value="vnd.wtv-green" /> <xs:enumeration value="vnd.wtv-y" /> <xs:enumeration value="vnd.wtv-yellow" /> <xs:enumeration value="vnd.wtv-b" /> <xs:enumeration value="vnd.wtv-blue" /> <xs:enumeration value="vnd.wtv-sel" /> <xs:enumeration value="vnd.wtv-p" /> <xs:enumeration value="vnd.wtv-m" /> <xs:enumeration value="vnd.wtv-text" /> <xs:enumeration value="vnd.wtv-i" /> <xs:enumeration value="vnd.wtv-0" /> <xs:enumeration value="vnd.wtv-1" /> <xs:enumeration value="vnd.wtv-2" /> <xs:enumeration value="vnd.wtv-3" /> <xs:enumeration value="vnd.wtv-4" /> <xs:enumeration value="vnd.wtv-5" /> <xs:enumeration value="vnd.wtv-6" /> <xs:enumeration value="vnd.wtv-7" /> <xs:enumeration value="vnd.wtv-8" /> <xs:enumeration value="vnd.wtv-9" /> <xs:enumeration value="vnd.wtv-up" /> <xs:enumeration value="vnd.wtv-down" /> <xs:enumeration value="vnd.wtv-left" /> <xs:enumeration value="vnd.wtv-right" /> <xs:enumeration value="vnd.wtv-home" /> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>					

	<pre> </xs:simpleType> </xs:attribute> <xs:attribute name="label" type="xs:string" use="optional"/> <xs:attribute name="title" type="xs:string" use="optional"/> <xs:attribute name="font" type="xs:string" use="optional"/> <xs:attribute name="aacolor" type="xs:string" use="optional"/> <xs:attribute name="pos" type="xs:string" use="optional"/> <xs:attribute name="bglocalsrc" type="xs:string" use="optional"/> <xs:attribute name="icon" type="xs:string" use="optional"/> <xs:attribute name="bgoffset" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>
--	---

The popupdo element is used to aggregate one or more dos into a popup menu.

The popupdo element is a special instance of wtvml:do, and has identical attributes and characteristics, with these exceptions.

popupdo has a single fixed action to launch a popup menu.

popupdo can only exist in modeset.xml files as a child element of mode.

The popupdo element can appear in the main menu, or can be displayed at any onscreen x,y coordinate in the same way as a floating do.

A popupdo must always have a "type" attribute definition. Any other wtvml:dos with the same "type" definition will be aggregated into the popup menu, except where those dos are inline.

Popup menus are context sensitive. A do will only be a candidate for inclusion in a popup menu if it is defined in the same mode as the popupdo, or if it is are defined in the current card and deck.

6.3.19 Popupmenu element

diagram					
namespace	http://waptv.com/xsd/modeset				
children	modeset:tr				
used by	element popupdo				
attributes	Name	Type	Use	Default	Fixed
	size	xs:string	optional		
	minsize	xs:string	optional		
	maxsize	xs:string	optional		
	ordered	xs:boolean	optional	false	
	rowgap	xs:string	optional		
	bgcolor	xs:string	optional		
	bgradius	xs:string	optional		
	borderwidth	xs:string	optional		
	bordercolor	xs:string	optional		
	hpad	xs:string	optional		
	vpad	xs:string	optional		
	hspace	xs:string	optional		
	vspace	xs:string	optional		
	align	xs:NMTOKEN	optional	tl	
	pos	xs:string	required		
	pivot	xs:NMTOKEN	optional		
source	<pre> <xs:element name="popupmenu"> <xs:annotation> <xs:appinfo> <comp:element-def name="popupmenu" bytecode="107"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:all minOccurs="0"> <xs:element ref="modeset:tr" minOccurs="0"/> </xs:all> <xs:attribute name="size" type="xs:string" use="optional"/> <xs:attribute name="minsize" type="xs:string" use="optional"/> <xs:attribute name="maxsize" type="xs:string" use="optional"/> </pre>				

```

<xs:attribute name="ordered" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="rowgap" type="xs:string" use="optional"/>
<xs:attribute name="bgcolor" type="xs:string" use="optional"/>
<xs:attribute name="bgradius" type="xs:string" use="optional"/>
<xs:attribute name="borderwidth" type="xs:string" use="optional"/>
<xs:attribute name="bordercolor" type="xs:string" use="optional"/>
<xs:attribute name="hpad" type="xs:string" use="optional"/>
<xs:attribute name="vpad" type="xs:string" use="optional"/>
<xs:attribute name="hspace" type="xs:string" use="optional"/>
<xs:attribute name="vspace" type="xs:string" use="optional"/>
<xs:attribute name="align" use="optional" default="tl">
  <xs:simpleType>
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="tl"/>
      <xs:enumeration value="tc"/>
      <xs:enumeration value="tr"/>
      <xs:enumeration value="ml"/>
      <xs:enumeration value="mc"/>
      <xs:enumeration value="mr"/>
      <xs:enumeration value="bl"/>
      <xs:enumeration value="bc"/>
      <xs:enumeration value="br"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="pos" type="xs:string" use="required"/>
<xs:attribute name="pivot" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="tl"/>
      <xs:enumeration value="tr"/>
      <xs:enumeration value="tc"/>
      <xs:enumeration value="ml"/>
      <xs:enumeration value="mr"/>
      <xs:enumeration value="mc"/>
      <xs:enumeration value="bl"/>
      <xs:enumeration value="br"/>
      <xs:enumeration value="bc"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

The popupmenu element describes the physical characteristics of a popup menu.

6.3.19.1 Pos attribute

pos="x,y"

The pos attributes defines the point to which the popupmenu will be anchored.

6.3.19.2 Pivot attribute

pivot="(tl|tm|tr|cl|cm|cr|bl|bm|br)"

The pivot attribute specifies the position of the popup menu relative to the point defined by the pos attribute.

For example, a pivot definition of "tl" will draw the popupmenu above, and to the left of the point defined by the pos attribute, so that the bottom right corner of the popup menu will be anchored to the point defined by the pos attribute.

Default value for pivot is "tl".

6.3.19.3 Size attribute

size="n"

size="n,m"

The size attribute specifies the size of the panel.

When the size value defined is a single positive integer, it specifies the number of dos that will be simultaneously visible in the popupmenu.

In this case, the total height of the popupmenu will be the cumulative height of each individual do. Note that any background images that may also be defined for each do are not included in the height calculation. Service authors are advised that where "bglocalsrc" images are to be used with popup menu dos, care should be taken to ensure the height of the background image matches the height of the do.

When the size value defined is a comma separated pair of positive integers, it specifies in pixels the exact size of the popupmenu.

In this case, the first integer specifies the width of the popup menu, and the second integer specifies the height. Either integer can be defined as "0" for unset.

Where a size attribute value is defined as "0", the microBrowser should automatically calculate the appropriate size of the popup menu.

6.3.19.4 Minsize attribute

The minsize attribute is an optional attribute used to specify in pixels the minimum size for the popupmenu. Minsize is defined as a pair of comma-separated positive integers.

The first integer specifies the minimum width of the popup menu, and the second integer specifies the minimum height. Either integer can be defined as "0" for unset.

6.3.19.5 Maxsize attribute

The maxsize attribute is an optional attribute used to specify in pixels the maximum size for the popupmenu. Maxsize is defined as a pair of comma-separated positive integers.

The first integer specifies the maximum width of the popupmenu, and the second integer specifies the maximum height. Either integer can be defined as "0" for unset.

6.3.19.6 Align attribute


align="(tl|tm|tr|cl|cm|cr|bl|bm|br)"

The align attribute specifies alignment of the underlying table control within the menu area.

6.3.19.7 Example

```
<popupdo pos="200,200" id="redpop" name="Red" type="vnd.wtv-r" label="Red">
  <popupmenu pos="400,30" hpad="20" vpad="5" rowgap="20" size="0,300" align="mc"/>
</popupdo>
<popupdo pos="300,200" id="yellowpop" name="Yellow" type="vnd.wtv-r" label="Yellow">
  <popupmenu pos="200,30" hpad="20" vpad="5" rowgap="20" size="0,300" align="mc"/>
</popupdo>
```

6.3.20 Scroll element

diagram					
namespace	http://waptv.com/xsd/modeset				
used by	elements browser dialog input optgroup option select				
attributes	Name	Type	Use	Default	Fixed
	fgcolor	xs:string	optional		
	fgfocused	xs:string	optional		
	bgcolor	xs:string	optional		
	bgfocused	xs:string	optional		
source	<pre> <xs:element name="scroll"> <xs:annotation> <xs:documentation>Defines default characteristics of scrollbar</xs:documentation> <xs:appinfo> <comp:element-def name="scroll" bytecode="0e"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="fgcolor" type="xs:string" use="optional"/> <xs:attribute name="fgfocused" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bgfocused" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

The scroll element defines the default characteristics of the scroll bar.

6.3.20.1 Fgcolor attribute

```
fgcolor="(#rrggbb|transp)"
```

The fgcolor attribute specifies the foreground colour of the scrollbar.

6.3.20.2 Fgfocused attribute

```
fgfocused="(#rrggbb|transp)"
```

The fgfocused attribute defines the foreground colour of the scroll bar when it has focus.

6.3.20.3 Bgcolor attribute

```
bgcolor="(#rrggbb|transp)"
```

The bgcolor attribute defines the background colour of the scrollbar.

6.3.20.4 Bgfocused attribute

```
bgfocused="(#rrggbb|transp)"
```

The bgfocused attribute defines the background colour of the scrollbar when it has focus.

6.3.20.5 Example

```

<browser visible='false' rect='180,320,660,502' bgcolor='' bordercolor='#0b146f'
borderfocused='#f0f0f0' >
  <scroll bgcolor='#070847' bgfocused='#070847' fgcolor='#39617b' fgfocused='#5a9cad' />
</browser>

<select bgcolor='#0d107a' bordercolor='#0d107a' borderfocused='#f0f0f0' >
  <scroll bgcolor='#0d107a' bgfocused='#0d107a' fgcolor='#39617b' fgfocused='#5a9cad' />
</select>

```


```

<optgroup bgcolor='#0d107a' bordercolor='#0d107a' borderfocused='#f0f0f0' >
  <scroll bgcolor='#0d107a' bgfocused='#0d107a' fgcolor='#39617b' fgfocused='#5a9cad' />
</optgroup >

<input bgcolor='#0d107a' cursorcolor='#58969c' bordercolor='#0d107a' borderfocused='#f0f0f0' >
  <scroll bgcolor='#xxxxx' bgfocused='#xxxxxx' fgcolor='#39617b' fgfocused='#5a9cad' />
</input>
</browser>

```

6.3.21 Select element

diagram					
namespace	http://waptv.com/xsd/modeset				
children	modeset:scroll				
used by	elements browser dialog				
attributes	Name	Type	Use	Default	Fixed
	bgcolor	xs:string	optional		
	bordercolor	xs:string	optional		
	borderfocused	xs:string	optional		
	style	xs:NMTOKEN	optional	normal	
	bgradius	xs:string	optional		
	borderwidth	xs:string	optional		
	localsrc	xs:string	optional		
	pivot	xs:NMTOKEN	optional	tl	
source	<pre> <xs:element name="select"> <xs:annotation> <xs:appinfo> <comp:element-def name="select" bytecode="37"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="modeset:scroll" minOccurs="0"/> </xs:sequence> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderfocused" type="xs:string" use="optional"/> <xs:attribute name="style" use="optional" default="normal"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="normal"/> <xs:enumeration value="autoup"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> <xs:attribute name="localsrc" type="xs:string" use="optional"/> <xs:attribute name="pivot" use="optional" default="tl"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="tl"/> <xs:enumeration value="tr"/> <xs:enumeration value="tc"/> <xs:enumeration value="ml"/> <xs:enumeration value="mr"/> <xs:enumeration value="mc"/> <xs:enumeration value="bl"/> <xs:enumeration value="br"/> <xs:enumeration value="bc"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </pre>				

The select element lets users pick from a list of options. Each option is specified by an option element. Option elements may be organized into hierarchical groups using the optgroup element.

6.3.21.1 Bgcolor attribute

```
bgcolor="#rrggbb|transp)"
```

The bgcolor attribute defines the background colour of the select list.

6.3.21.2 Bordercolor attribute

```
bordercolor="#rrggbb|transp)"
```

The bordercolor attribute defines in RGB values the colour for the select list border.

6.3.21.3 Borderfocused attribute

```
borderfocused="#rrggbb|transp)"
```

The borderfocused attribute defines in RGB values the colour for the select list border when the select has focus.

6.3.21.4 Style attribute

```
style="normal|autoup"
```

The style attribute specifies the behaviour of the select object. If this attribute is unspecified, or is specified as "normal" a select object will only open to reveal its select list when the user has "clicked" on the focused object.

When the style attribute is defined as an autoup, the select list will open on focus, without requiring the user to click on the object to open it.

The style attribute of the select element can be specified as part of a mode. If it is set to autoup at the mode level, then all select objects in cards referring to this mode will display the autoup behaviour.

A mode level autoup behaviour can be overridden at the card level.

6.3.21.5 Bgradius attribute

```
bgradius="10"
```

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the select object.

6.3.21.6 Borderwidth attribute

```
borderwidth=2
```

The borderwidth attribute defines in pixels the width of the object's border. Default value is two pixels.

6.3.21.7 Pivot attribute

```
pivot="(tl|tm|tr|cl|cm|cr|bl|bm|br)"
```

The pivot attribute specifies which corner or side of the select object will anchor the select list when it pops up.

Default behaviour is for the top left corner on the popupmenu to be anchored to the point defined by the pos attribute.

6.3.21.8 Example

```
<browser visible='false' rect='180,320,660,502' bgcolor='' bordercolor='#0b146f'
borderfocused='#f0f0f0' >
  <scroll bgcolor='#070847' bgfocused='#070847' fgcolor='#39617b' fgfocused='#5a9cad' />
<select bgcolor='#0d107a' bordercolor='#0d107a' borderfocused='#f0f0f0' style='autoup'>
  <scroll bgcolor='#0d107a' bgfocused='#0d107a' fgcolor='#39617b' fgfocused='#5a9cad' />
</select>
```

```

<optgroup bgcolor='#0d107a' bordercolor='#0d107a' borderfocused='#f0f0f0' >
  <scroll bgcolor='#0d107a' bgfocused='#0d107a' fgcolor='#39617b' fgfocused='#5a9cad' />
</optgroup >

<input bgcolor='#0d107a' cursorcolor='#58969c' bordercolor='#0d107a' borderfocused='#f0f0f0' >
  <scroll bgcolor='#xxxxx' bgfocused='#xxxxxx' fgcolor='#39617b' fgfocused='#5a9cad' />
</input>
</browser>

```

6.3.22 Optgroup element

diagram	<p>Defines default characteristics of Optgroups</p> <p>Defines default characteristics of scrollbar</p>				
namespace	http://waptv.com/xsd/modeset				
children	modeset:scroll				
used by	elements browser dialog				
attributes	Name	Type	Use	Default	Fixed
	bgcolor	xs:string	optional		
	bordercolor	xs:string	optional		
	borderfocused	xs:string	optional		
	bgradius	xs:string	optional		
	borderwidth	xs:string	optional		
source	<pre> <xs:element name="optgroup"> <xs:annotation> <xs:appinfo> <comp:element-def name="optgroup" bytecode="34"/> </xs:appinfo> <xs:documentation>Defines default characteristics of Optgroups</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="modeset:scroll" minOccurs="0"/> </xs:sequence> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderfocused" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

The optgroup element is used to define hierarchical groupings of options within a select list.

6.3.22.1 Bgcolor attribute

```
bgcolor="(#rrggbb|transp)"
```

The bgcolor attribute defines the background colour of the optgroup.

6.3.22.2 Bordercolor attribute

```
bordercolor="(#rrggbb|transp)"
```

The bordercolor attribute defines in RGB values the colour for the optgroup.

6.3.22.3 Borderfocused attribute

```
borderfocused="(#rrggbb|transp)"
```

The borderfocused attribute defines in RGB values the colour for the optgroup border when the optgroup has focus.

6.3.22.4 Bgradius attribute

bgradius="10"

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the menu object.

6.3.22.5 Borderwidth attribute

borderwidth="10"

The borderwidth attribute defines in pixels the width of the object's border. Default value is two pixels.

6.3.22.6 Example

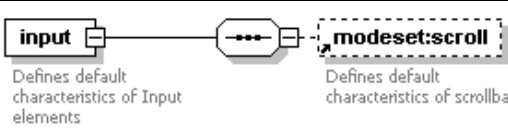
```
<browser visible='false' rect='180,320,660,502' bgcolor='' bordercolor='#0b146f'
borderfocused='#f0f0f0' >
  <scroll bgcolor='#070847' bgfocused='#070847' fgcolor='#39617b' fgfocused='#5a9cad' />

<select bgcolor='#0d107a' bordercolor='#0d107a' borderfocused='#f0f0f0' >
  <scroll bgcolor='#0d107a' bgfocused='#0d107a' fgcolor='#39617b' fgfocused='#5a9cad' />
</select>

<optgroup bgcolor='#0d107a' bordercolor='#0d107a' borderfocused='#f0f0f0' >
  <scroll bgcolor='#0d107a' bgfocused='#0d107a' fgcolor='#39617b' fgfocused='#5a9cad' />
</ optgroup >

<input bgcolor='#0d107a' cursorcolor='#58969c' bordercolor='#0d107a' borderfocused='#f0f0f0' >
  <scroll bgcolor='#xxxxx' bgfocused='#xxxxxx' fgcolor='#39617b' fgfocused='#5a9cad' />
</input>
</browser>
```

6.3.23 Input element

diagram					
namespace	http://waptv.com/xsd/modeset				
children	modeset:scroll				
used by	elements browser dialog				
attributes	Name	Type	Use	Default	Fixed
	cursorcolor	xs:string	optional		
	bgcolor	xs:string	optional		
	borderwidth	xs:string	optional		
	bordercolor	xs:string	optional		
	borderfocused	xs:string	optional		
	borderunfocused	xs:string		optional	
	bgradius	xs:string	optional		

source	<pre> <xs:element name="input"> <xs:annotation> <xs:documentation>Defines default characteristics of Input elements</xs:documentation> <xs:appinfo> <comp:element-def name="input" bytecode="2f"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="modeset:scroll" minOccurs="0"/> </xs:sequence> <xs:attribute name="cursorcolor" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderfocused" type="xs:string" use="optional"/> <xs:attribute name="borderunfocused" type="xs:string" use="optional"/> <xs:attribute name="bgradius" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>
---------------	---

The input element defines the default appearance of text input fields. The constraints on input (e.g. numeric, password, text etc.) are defined in the WML.

6.3.23.1 Cursorcolor attribute

cursorcolor="(#rrggbb|transp)"

The cursorcolor attribute defines the colour of the cursor when in an input field.

6.3.23.2 Bgcolor attribute

bgcolor="(#rrggbb|transp)"

The bgcolor attribute defines the background colour of the input field.

6.3.23.3 Borderwidth attribute

borderwidth="10"

The borderwidth attribute defines in pixels the width of the input object's border.

6.3.23.4 Bordercolor attribute

bordercolor="(#rrggbb|transp)"

The bordercolor attribute defines in RGB values the colour for the input field.

6.3.23.5 Borderfocused attribute

borderfocused="(#rrggbb|transp)"

The borderfocused attribute defines in RGB values the colour for the input field border when the input field has focus.

6.3.23.6 Bgradius attribute

bgradius="10"

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the menu object.

6.3.23.7 Borderwidth attribute

borderwidth="10"

The borderwidth attribute defines in pixels the width of the object's border. Default value is two pixels.

6.3.23.8 Example

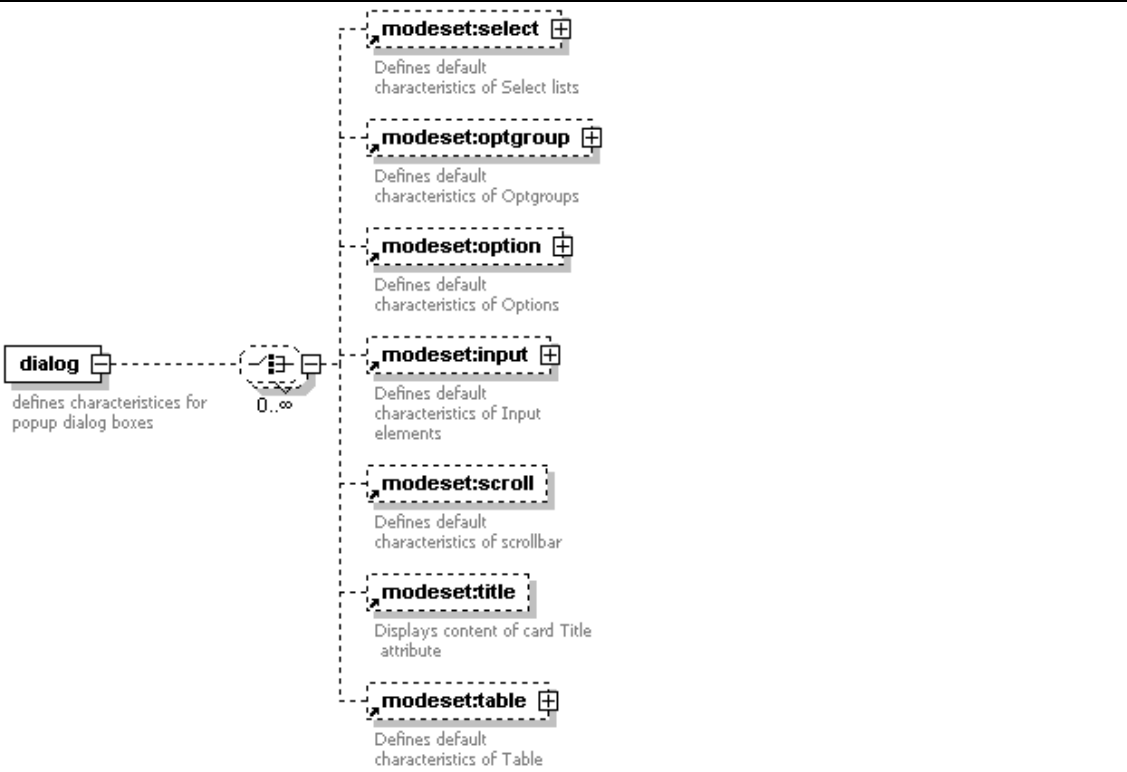
```
<browser visible='false' rect='180,320,660,502' bgcolor='' bordercolor='#0b146f'
borderfocused='#f0f0f0' >
  <scroll bgcolor='#070847' bgfocused='#070847' fgcolor='#39617b' fgfocused='#5a9cad' />

  <select bgcolor='#0d107a' bordercolor='#0d107a' borderfocused='#f0f0f0' >
    <scroll bgcolor='#0d107a' bgfocused='#0d107a' fgcolor='#39617b' fgfocused='#5a9cad' />
  </select>

  <optgroup bgcolor='#0d107a' bordercolor='#0d107a' borderfocused='#f0f0f0' >
    <scroll bgcolor='#0d107a' bgfocused='#0d107a' fgcolor='#39617b' fgfocused='#5a9cad' />
  </optgroup >

  <input bgcolor='#0d107a' cursorcolor='#58969c' bordercolor='#0d107a' borderfocused='#f0f0f0' >
    <scroll bgcolor='#xxxxx' bgfocused='#xxxxx' fgcolor='#39617b' fgfocused='#5a9cad' />
  </input>
</browser>
```

6.3.24 Dialog element

<p>diagram</p> 	<p>dialog defines characteristics for popup dialog boxes</p> <p>0..∞</p> <p>modeset:select Defines default characteristics of Select lists</p> <p>modeset:optgroup Defines default characteristics of Optgroups</p> <p>modeset:option Defines default characteristics of Options</p> <p>modeset:input Defines default characteristics of Input elements</p> <p>modeset:scroll Defines default characteristics of scrollbar</p> <p>modeset:title Displays content of card Title attribute</p> <p>modeset:table Defines default characteristics of Table</p>
namespace	http://waptv.com/xsd/modeset
children	modeset:select modeset:optgroup modeset:option modeset:input modeset:scroll modeset:title modeset:table
used by	element mode

attributes	Name	Type	Use	Default	Fixed dialog
	id	xs:string	optional		
	visible	xs:boolean	optional	true	
	rect	xs:string	required		
	aacolor	xs:string	optional		
	bgcolor	xs:string	optional		
	bordercolor	xs:string	optional		
	borderfocused	xs:string	optional		
	type	xs:string	optional		
	font	xs:string	optional		
	borderwidth	xs:integer	optional		
	bgradius	xs:integer	optional		
source	<pre> <xs:element name="dialog"> <xs:annotation> <xs:documentation>defines characteristics for popup dialog boxes</xs:documentation> <xs:appinfo> <comp:element-def name="dialog" bytecode="09"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="modeset:select" minOccurs="0"/> <xs:element ref="modeset:optgroup" minOccurs="0"/> <xs:element ref="modeset:option" minOccurs="0"/> <xs:element ref="modeset:input" minOccurs="0"/> <xs:element ref="modeset:scroll" minOccurs="0"/> <xs:element ref="modeset:title" minOccurs="0"/> <xs:element ref="modeset:table" minOccurs="0"/> </xs:choice> <xs:attribute name="id" type="xs:string" use="optional" fixed="dialog"/> <xs:attribute name="visible" type="xs:boolean" use="optional" default="true"/> <xs:attribute name="rect" type="xs:string" use="required"/> <xs:attribute name="aacolor" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderfocused" type="xs:string" use="optional"/> <xs:attribute name="type" type="xs:string" use="optional"/> <xs:attribute name="font" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:integer" use="optional"/> <xs:attribute name="bgradius" type="xs:integer" use="optional"/> </xs:complexType> </xs:element> </pre>				

The dialog element controls the display of popup dialogs. There must be a single dialog instance defined, and it will be allocated the id dialog.

The dialog is used to display errors or other information to the user.

Because a dialog can contain full WML markup, it is necessary to set the default characteristics for WML elements (e.g. select, optgroup, input, scroll) as with a browser element. To display a WML deck in the popup dialog, use the target attribute of a go, with the value "dialog".

EXAMPLE: <go href='popup.wml' target='dialog' />.

6.3.24.1 Id attribute

id="dialog"

The id attribute defines a unique id for the dialog. This attribute can be manipulated from the WML content using the setattr element.

6.3.24.2 Visible element

visible="true"

Determines whether the dialog will be visible onscreen or not.

6.3.24.3 Rect attribute

```
rect=x1,y1,x2,y2
```

The rect attribute defines where the dialog pane is to be drawn on the screen.

6.3.24.4 Aacolor attribute

```
aacolor="(#rrggbb|transp)"
```

The aacolor attribute defines the colour that all the text elements in the dialog will be anti-aliased against.

6.3.24.5 Bgcolor attribute

```
bgcolor="(#rrggbb|transp)"
```

The bgcolor attribute defines the background colour of the dialog. If bgcolor is defined, the text in the dialog will also be anti-aliased against this colour. If it is not defined, no box will be drawn and the value of aacolor will be used for anti-aliasing.

6.3.24.6 Bordercolor attribute

```
bordercolor="(#rrggbb|transp)"
```

The bordercolor attribute defines in RGB values the colour for the dialog box.

6.3.24.7 Borderfocused attribute

```
borderfocused="(#rrggbb|transp)"
```

The borderfocused attribute defines in RGB values the colour for the dialog box border when the dialog box has focus.

6.3.24.8 Bgradius attribute

```
bgradius="10"
```

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the dialog object.

6.3.24.9 Borderwidth attribute

```
borderwidth="10"
```

The borderwidth attribute defines in pixels the width of the object's border.

6.3.24.10 Example

```
<dialog rect='140,380,580,520' bgcolor='#070847' borderfocused='#f0f0f0'>
  <scroll bgcolor='#070847' bgfocused='#070847' fgcolor='#39617b' fgfocused='#5a9cad' />
</dialog>
```

6.3.25 Browser element

<p>diagram</p> <p>browser Defines characteristics for browser pane(s). Must be minimum of one browser pane defined per mode.</p> <p>0..∞</p> <p>modeset:select Defines default characteristics of Select lists</p> <p>modeset:optgroup Defines default characteristics of Optgroups</p> <p>modeset:option Defines default characteristics of Options</p> <p>modeset:input Defines default characteristics of Input elements</p> <p>modeset:table Defines default characteristics of Table</p> <p>modeset:scroll Defines default characteristics of scrollbar</p> <p>modeset:title Displays content of card Title attribute</p>																																																								
namespace	http://waptv.com/xsd/modeset																																																							
children	modeset:select modeset:optgroup modeset:option modeset:input modeset:table modeset:scroll modeset:title																																																							
used by	element mode																																																							
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>xs:string</td> <td>optional</td> <td>main</td> <td></td> </tr> <tr> <td>type</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>normal</td> <td></td> </tr> <tr> <td>visible</td> <td>xs:NMTOKEN</td> <td>optional</td> <td>true</td> <td></td> </tr> <tr> <td>rect</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>aacolor</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bgcolor</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bordercolor</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>borderfocused</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>borderwidth</td> <td>xs:integer</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>bgradius</td> <td>xs:integer</td> <td>optional</td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	id	xs:string	optional	main		type	xs:NMTOKEN	optional	normal		visible	xs:NMTOKEN	optional	true		rect	xs:string	required			aacolor	xs:string	optional			bgcolor	xs:string	optional			bordercolor	xs:string	optional			borderfocused	xs:string	optional			borderwidth	xs:integer	optional			bgradius	xs:integer	optional		
Name	Type	Use	Default	Fixed																																																				
id	xs:string	optional	main																																																					
type	xs:NMTOKEN	optional	normal																																																					
visible	xs:NMTOKEN	optional	true																																																					
rect	xs:string	required																																																						
aacolor	xs:string	optional																																																						
bgcolor	xs:string	optional																																																						
bordercolor	xs:string	optional																																																						
borderfocused	xs:string	optional																																																						
borderwidth	xs:integer	optional																																																						
bgradius	xs:integer	optional																																																						
source	<pre> <xs:element name="browser"> <xs:annotation> <xs:documentation>Defines characteristics for browser pane(s). Must be minimum of one browser pane defined per mode.</xs:documentation> <xs:appinfo> <comp:element-def name="browser" bytecode="08"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="modeset:select" minOccurs="0"/> <xs:element ref="modeset:optgroup" minOccurs="0"/> <xs:element ref="modeset:option" minOccurs="0"/> <xs:element ref="modeset:input" minOccurs="0"/> <xs:element ref="modeset:table" minOccurs="0"/> <xs:element ref="modeset:scroll" minOccurs="0"/> <xs:element ref="modeset:title" minOccurs="0"/> </xs:choice> <xs:attribute name="id" type="xs:string" use="optional" default="main"/> <xs:attribute name="type" use="optional" default="normal"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="normal"/> <xs:enumeration value="popup"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </pre>																																																							

	<pre> </xs:attribute> <xs:attribute name="visible" use="optional" default="true"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="true"/> <xs:enumeration value="false"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="rect" type="xs:string" use="required"/> <xs:attribute name="aacolor" type="xs:string" use="optional"/> <xs:attribute name="bgcolor" type="xs:string" use="optional"/> <xs:attribute name="bordercolor" type="xs:string" use="optional"/> <xs:attribute name="borderfocused" type="xs:string" use="optional"/> <xs:attribute name="borderwidth" type="xs:integer" use="optional"/> <xs:attribute name="bgradius" type="xs:integer" use="optional"/> </xs:complexType> </xs:element> </pre>
--	--

The browser element defines the characteristics of a browser pane.

The characteristics defined in the table subelement, and in its tr and td subelements define default table properties for WML displayed in the mode.

Attributes for all browser sub elements select, optgroup, input, scroll and title must be independently specified in the first mode of the modeset. These attributes will not be inherited from the browser.

The first browser instance will be the main browser and its id must be main. Subsidiary browsers may also exist within a single mode. These may have any id specified by the application author.

Changes between modes can only be initiated from WML loaded in the main browser.

All attributes of browser subelements select, optgroup, input, scroll, title and table will persist across all browsers. These subelements are defined in the main browser, but are shared across all browsers and will have the same attributes in all browsers.

Attributes of the browser object itself can be individually specified for each browser.

6.3.25.1 Id attribute

```
id="main"
```

The id attribute defines a unique id for the browser.

The primary browser must have the id main. Subsidiary browsers may have any id defined by the application author.

6.3.25.2 Type attribute

```
type="(normal|popup)"
```

The type attribute defines whether the Browser should be a popup or a normal browser pane. If unspecified, the browser type will default to normal.

Where defined as a popup, the Browser should be modal. This means that any accelerated dos in any underlying browser or mode will be disabled while the popup is active.

6.3.25.3 Visible attribute

```
visible="true"
```

Determines whether the Browser should be visible onscreen or not.

6.3.25.4 Rect attribute

```
rect=x1,y1,x2,y2
```

The rect attribute defines where the browser pane is to be drawn on the screen.

6.3.25.5 Aacolor attribute

```
aacolor="(#rrggbb|transp)"
```

The aacolor attribute defines the colour that all the text elements in the Browser should be anti-aliased against.

6.3.25.6 Bgcolor attribute

```
bgcolor="(#rrggbb|transp)"
```

The bgcolor attribute defines the background colour of the browser. If bgcolor is defined, the text in the dialog will also be anti-aliased against this colour. If it is not defined, no box will be drawn and the value of aacolor will be used for anti-aliasing.

6.3.25.7 Bordercolor attribute

```
bordercolor="(#rrggbb|transp)"
```

The bordercolor attribute defines in RGB values the colour for the browser.

6.3.25.8 Borderfocused attribute

```
borderfocused="(#rrggbb|transp)"
```

The borderfocused attribute defines in RGB values the colour for the browser border when the browser has focus.

6.3.25.9 Bgradius attribute

```
bgradius="10"
```

The bgradius attribute defines in pixels the radius of a curve used to "round off" the corners of the browser object.

6.3.25.10 Borderwidth attribute

```
borderwidth="10"
```

The borderwidth attribute defines in pixels the width of the object's border.

6.3.25.11 Example

```
<?xml version="1.0"?>

<modeset fontsrc='http://menu.waptv.co.uk/fontset.wml' localsrc='http://resources/ires.res#1000' >

  <mode id='normal' bgimage='http://menu.waptv.co.uk/skybg.mpg' >
    <av mode='window' rect='336,48,656,305' hscale='1' vscale='1' toggle='27' />
    <resume rect='500,48,656,200' localsrc='resume' />
    <status rect='100,75,200,175' />
    <hint rect='180,504,660,532' align='bl' font='hint' bgcolor='#0b146f' />
    <callrate rect='180,78,320,128' align='bl' font='callrate' bgcolor='#07084f' />
    <menu font='menu' rect='0,175,158,528' align='bl' rowgap='14' aacolor='#ff0000' >
      <do localsrc='menubox' pos='55,0' font='menu' />
    </menu>
    <dialog font='dialog' rect='140,270,580,510' bgcolor='#070847' borderfocused='#f0f0f0' />
    <browser rect='180,320,660,502' bgcolor='#0b146f' bordercolor='#0b146f' borderfocused='#f0f0f0'
  >
      <scroll bgcolor='#07084f' bgfocused='#07084f' fgcolor='#39617b' fgfocused='#5a9cad' />
```

```

<title rect='180,128,320,314' font='title' aacolor='#ff0000' align='bl' bordercolor='#ffffff'
bgradius='10' borderwidth='2' vpad='10' hpad='15' />
<select bgcolor='#07084f' bordercolor='#070847' borderfocused='#f0f0f0' select='autoup' >
  <scroll bgcolor='#07084f' bgfocused='#07084f' fgcolor='#39617b' fgfocused='#5a9cad' />
</select>
<optgroup bgcolor='#07084f' bordercolor='#070847' borderfocused='#f0f0f0' >
  <scroll bgcolor='#07084f' bgfocused='#07084f' fgcolor='#39617b' fgfocused='#5a9cad' />
</optgroup>
<option bgcolor='#07084f' bordercolor='#070847' borderfocused='#f0f0f0' >
  <scroll bgcolor='#07084f' bgfocused='#07084f' fgcolor='#39617b' fgfocused='#5a9cad' />
</option>
<input bgcolor='#07084f' cursor='#58969c' bordercolor='#070847' borderfocused='#f0f0f0' >
  <scroll bgcolor='#07084f' bgfocused='#07084f' fgcolor='#39617b' fgfocused='#5a9cad' />
</input>
</browser>
<browser id='second' rect='180,120,660,310' bgcolor='#0b146f' bordercolor='#0b146f'
borderfocused='#f0f0f0' />
<browser id='pop' type='popup' rect='180,120,660,310' bgcolor='#ff0000' bordercolor='#0b146f'
borderfocused='#f0f0f0' />
<img id='usl' rect='180,49,280,149' />
<text id="text1" rect='180,400,320,450' font='title' />
<img id="img1" rect='280,400,380,500' />
<img id="img2" rect='380,400,480,500' />

</mode>

<mode id='full' >
  <av mode='blank' />
  <browser rect='180,120,660,502' />
  <hint rect='180,104,660,132' />
</mode>
</modeset>

```

6.3.26 Mode element

<p>diagram</p>																					
<p>namespace</p>	<p>http://waptv.com/xsd/modeset</p>																				
<p>children</p>	<p>modeset:av modeset:menu modeset:status modeset:resume modeset:hint modeset:callrate modeset:dialog modeset:browser modeset:clock wtvml:do wtvml:precache modeset:text modeset:img modeset:box</p>																				
<p>used by</p>	<p>element modeset</p>																				
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>bgimage</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> </tr> <tr> <td>translucency</td> <td>xs:integer</td> <td>optional</td> <td>128</td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	id	xs:string	required			bgimage	xs:string	optional			translucency	xs:integer	optional	128	
Name	Type	Use	Default	Fixed																	
id	xs:string	required																			
bgimage	xs:string	optional																			
translucency	xs:integer	optional	128																		

source	<pre> <xs:element name="mode"> <xs:annotation> <xs:appinfo> <comp:element-def name="mode" bytecode="06" /> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element ref="modeset:av" minOccurs="0" /> <xs:element ref="modeset:menu" minOccurs="0" /> <xs:element ref="modeset:status" /> <xs:element ref="modeset:resume" minOccurs="0" /> <xs:element ref="modeset:hint" minOccurs="0" /> <xs:element ref="modeset:callrate" /> <xs:element ref="modeset:dialog" minOccurs="0" /> <xs:element ref="modeset:browser" maxOccurs="unbounded" /> <xs:element ref="modeset:clock" minOccurs="0" maxOccurs="unbounded" /> <xs:element ref="wtvml:do" minOccurs="0" maxOccurs="unbounded" /> <xs:element ref="wtvml:precache" minOccurs="0" maxOccurs="unbounded" /> <xs:element ref="modeset:text" minOccurs="0" /> <xs:element ref="modeset:img" minOccurs="0" maxOccurs="unbounded" /> <xs:element ref="modeset:box" minOccurs="0" maxOccurs="unbounded" /> </xs:choice> <xs:attribute name="id" type="xs:string" use="required" /> <xs:attribute name="bgimage" type="xs:string" use="optional" /> <xs:attribute name="translucency" type="xs:integer" use="optional" default="128" /> </xs:complexType> </xs:element> </pre>
---------------	---

The mode element is an "umbrella" within which a set of specifications for the location and appearance characteristics of all built-in onscreen objects can be defined. A browser is a compulsory object within a mode.

Modes and modesets are defined in an XML document which can be referenced via a href from the WML content.

Application authors can change between modes within the current modeset using the card level mode attribute. This is a URL identifying the resource containing the current modeset element, and the fragment identifying the mode within the modeset. Where the fragment is not specified, the first mode in the modeset will be used by default.

Changes between modes can only be initiated from the main browser.

6.3.26.1 Id attribute

```
id="mymode"
```

The id attribute is used to define a unique id for the mode. This attribute allows authors to specify changes between modes from within the WML content.

6.3.26.2 Bgimage attribute

```
bgimage="URL"
```

The bgimage attribute specifies the MPEG background image to be loaded. If not specified, the background image is inherited from the previous mode.

6.3.26.3 Translucency attribute

```
translucency="128"
```

The translucency attribute of the mode element defines a global translucency value for all translucent objects to be rendered under this current mode.

Translucency is defined as a number with a value between 0 and 255, where 0 is opaque and 255 is transparent.

The global translucency value can also be defined by using setattr to manipulate wtv.mode.fragment in the WTVML browser document object model.

For example, to set a global translucency value using setattr:

- `<setattr name="wtv.mode.translucency" value="128"/>`

The global translucency setting can be applied to any individual OSD object instance through appending the alpha value "01" to the RGB value triplets.

For example, to apply the globally defined translucency value to a browser with background colour #0B146F:

- `<browser ... bgcolor="#0B146F01">`

Any onscreen object which can support RGB colour definition can also have the "01" alpha channel appended to its colour definition attribute values to toggle application of globally translucency value. This can be applied to bgcolor, bordercolor, borderfocused attributes for any object.

Where the global translucency value has not been set, the "01" alpha value channel applied to any OSD object will have no effect.

6.3.26.4 Example

```
<wml mode='layout.xml'>

<head>
  <meta http-equiv="Cache-Control" content="no-cache" />
</head>

<template>
  <do type='prev'>
    <prev/>
  </do>
</template>

<card id="football" title='Football Headlines' mode='#normal'>
The full path of the mode must be specified if it is not within the same directory as the WML page
on the web server.
<?xml version="1.0"?>

<modeset fontsrc='fontset.xml' >

<mode id='skyenq' bgimage='skybg.mpg' >
  <av mode='blank' />
  <menu font='menu' rect='440,0,650,132' align='br' rowgap='0' >
    <do pos='10,0' font='menu' bg='' visible="true"/>
  </menu>

  <browser rect='75,140,660,470' bgcolor='#FF9900' >
    <scroll bgcolor='#10105A' bgfocused='#10105A' fgcolor='#39617b' fgfocused='#5a9cad' />
    <input bgcolor='#10105A' borderwidth='' cursorcolor='#ebbe0f' bordercolor='#10105A'
borderunfocused='#10105A' borderfocused='#C8C8C8'>
      <scroll bgcolor='#10105A' bgfocused='#10105A' fgcolor='#39617b' fgfocused='#5a9cad' />
    </input>
    <title rect='75,65,660,132' font='title' bgcolor='' align='bl' visible="true"/>
  </browser>

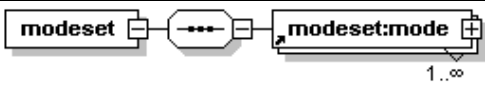
  <dialog font='dialog' rect='140,270,580,510' bgcolor='#070847' borderfocused='#f0f0f0' />
  <status rect='80,486,320,128' visible="true"/>
  <hint id='hint' rect='150,470,650,550' align='tl' font='hint' bgcolor='' visible="true"/>
  <callrate rect='150,510,440,540' align='bl' font='callrate' bgcolor='' visible="true" />

    <do id='v3' type='vnd.wtv-r' label='Home' focus='false'>
      <go href='http://skyenq.waptv.co.uk/index.wml' />
    </do>

</mode>

</modeset>
```

6.3.27 Moderset element

diagram																									
namespace	http://waptv.com/xsd/modeset																								
children	modeset:mode																								
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>fontsrc</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>localsrc</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>dosrc</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	fontsrc	xs:string	optional				localsrc	xs:string	optional				dosrc	xs:string	optional			
Name	Type	Use	Default	Fixed	Annotation																				
fontsrc	xs:string	optional																							
localsrc	xs:string	optional																							
dosrc	xs:string	optional																							
annotation	<pre> applInfo <comp:element-def name="modeset" bytecode="05" isroot="true"/> <comp:allowed-attribute name="xmlns" discard="true"/> <comp:allowed-attribute name="xmlns:comp" discard="true"/> <comp:allowed-attribute name="xmlns:wtml" discard="true"/> <comp:allowed-attribute name="xmlns:xsi" discard="true"/> <comp:allowed-attribute name="xsi:schemaLocation" discard="true"/> </pre>																								
source	<pre> <xs:element name="modeset"> <xs:annotation> <xs:appinfo> <comp:element-def name="modeset" bytecode="05" isroot="true"/> <comp:allowed-attribute name="xmlns" discard="true"/> <comp:allowed-attribute name="xmlns:comp" discard="true"/> <comp:allowed-attribute name="xmlns:wtml" discard="true"/> <comp:allowed-attribute name="xmlns:xsi" discard="true"/> <comp:allowed-attribute name="xsi:schemaLocation" discard="true"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="modeset:mode" maxOccurs="unbounded" /> </xs:sequence> <xs:attribute name="fontsrc" type="xs:string" use="optional" /> <xs:attribute name="localsrc" type="xs:string" use="optional" /> <xs:attribute name="dosrc" type="xs:string" use="optional" /> </xs:complexType> </xs:element> </pre>																								

A modeset element is a collection of modes. Attributes of elements within a modeset are cascading. This means that all characteristics of the previous mode will be inherited by a new mode unless explicitly overwritten.

Generally, a modeset should be designed so that there is an initial large mode that sets up all the screen objects, and other smaller modes within the set that perform any required incremental changes. For efficiency reasons, it is usually best to avoid reapplying the initial mode.

A new modeset can be loaded only by loading a new WML deck. Unlike modes, modesets are not cascading. The developer must redefine all desired elements and attributes when moving to a new modeset.

The modeset can also contain references to supporting XML files fontset.xml and doset.xml.

6.3.27.1 Fontsrc attribute

```
fontsrc="HREF"
```

The fontsrc attribute identifies the URL of the XML file which defines the font resource locations and the colour definitions used for fonts in this service.

6.3.27.2 Localsrc attribute

```
localsrc="HREF"
```

The localsrc attribute identifies the URL of the resource containing the BMP images used for service-specific images (such as specific navigation icons). This image set can use a maximum of 100 colours and must finish at colour 255.

The space from colour 96 to the start of the localsrc image set is used for dynamic allocation of colours.

6.3.27.3 Dosrc attribute

```
dosrc="HREF"
```

The dosrc attribute specifies the URL of the XML file which defines any mode-level dos to be included in the menu for this service.

6.3.27.4 Example

```
<?xml version="1.0"?>

<modeset fontsrc='fontset.xml' localsrc='http://resources/ires.res#1000' dosrc='doset.xml'>

  <mode id='normal' bgimage='skybg.mpg' >
    <av mode='window' rect='336,48,656,305' hscale='1' vscale='1' toggle='27' />
    <resume rect='500,48,656,200' localsrc='resume' />
    <status rect='100,75,200,175' />
    <hint rect='180,504,660,532' align='bl' font='hint' bgcolor='#0b146f' />
    <callrate rect='180,78,320,128' align='bl' font='callrate' bgcolor='#07084f' />
    <menu font='menu' rect='0,175,158,528' align='bl' rowgap='14' >
      <do localsrc='menubox' pos='55,0' font='menu' bgcolor='' />
    </menu>
    <dialog font='dialog' rect='140,270,580,510' bgcolor='#07084f' borderfocused='#f0f0f0' />

  <browser rect='180,320,660,502' bgcolor='#0b146f' bordercolor='#0b146f' borderfocused='#f0f0f0' >
    <scroll bgcolor='#07084f' bgfocused='#07084f' fgcolor='#39617b' fgfocused='#5a9cad' />
    <select bgcolor='#07084f' bordercolor='#07084f' borderfocused='#f0f0f0' >
      <scroll bgcolor='#07084f' bgfocused='#07084f' fgcolor='#39617b' fgfocused='#5a9cad' />
    </select>
    <optgroup bgcolor='#07084f' bordercolor='#07084f' borderfocused='#f0f0f0' >
      <scroll bgcolor='#07084f' bgfocused='#07084f' fgcolor='#39617b' fgfocused='#5a9cad' />
    </optgroup>
    <input bgcolor='#07084f' cursor='#58969c' bordercolor='#07084f' borderfocused='#f0f0f0' >
      <scroll bgcolor='#07084f' bgfocused='#07084f' fgcolor='#39617b' fgfocused='#5a9cad' />
    </input>
    <title rect='180,128,320,314' font='title' bgcolor='#ff0000' align='bl' />
  </browser>

  <img id='banner' />
  <img id='usl' rect='180,49,280,149' />
</mode>

  <mode id='full' >
    <av mode='blank' />
    <browser rect='180,120,660,502' />

  <do id='mode2' type='accept' label='mode do' >
    <refresh>
    <setattr name='wtv.mode' value='normal' />
    </refresh>
  </do>
</mode>

</modeset>
```

7 WTVML 6.3 Fontset specification

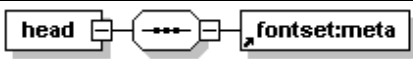
Fonts for any WTVML service can be defined outside the WTVML markup in a separate fonset XML file.

Fontsets are useful because they allow application authors to easily support uniformity of fonts to be used within families of interactive TV services, and they provide a mechanism to deliver more compact and optimized WTVML markup than would otherwise be possible.

7.1 Cache control

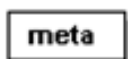
Cache behaviour for fontset files can be controlled through the meta element in the head of the modesetinstance.xml file, in the same way that cache behaviour for wtvml files can be defined in the meta element.

7.1.1 Head element

diagram	
namespace	http://waptv.com/xsd/fontset
children	fontset:meta
used by	element fontset
source	<pre><xs:element name="head"> <xs:complexType> <xs:sequence> <xs:element ref="fontset:meta" /> </xs:sequence> </xs:complexType> </xs:element></pre>

The head element can contain a single meta element, and has no attributes.

7.1.2 Meta element

diagram																			
namespace	http://waptv.com/xsd/fontset																		
used by	element head																		
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>http-equiv</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>content</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	http-equiv	xs:string	optional				content	xs:string	optional			
Name	Type	Use	Default	Fixed	Annotation														
http-equiv	xs:string	optional																	
content	xs:string	optional																	
source	<pre><xs:element name="meta"> <xs:complexType> <xs:attribute name="http-equiv" type="xs:string" use="optional" /> <xs:attribute name="content" type="xs:string" use="optional" /> </xs:complexType> </xs:element></pre>																		

7.1.2.1 Http-equiv attribute

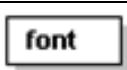
The http-equiv attribute provides a mechanism to set an http header, where the http-equiv value is the header name, and content is the header value. The network's transcoding server will convert an "http-equiv" meta value to an http response header before sending the file to the WTVML microbrowser browser.

7.1.2.2 Content attribute

When set to a value of "no-cache", the http-equiv meta value will instruct the microbrowser not to cache the fontsetinstance.xml file.

7.2 Fonts and fontsets

7.2.1 Font element

diagram					
namespace	http://waptv.com/xsd/fontset				
used by	element fontset				
attributes	Name	Type	Use	Default	Fixed
	id	xs:string	optional		
	src	xs:anyURI	required		
	size	xs:integer	optional		
	fgstatic	xs:string	optional		
	fgcolor	xs:string	optional		
	fgfocused	xs:string	optional		
	fgactivated	xs:string	optional		
source	<pre> <xs:element name="font"> <xs:annotation> <xs:appinfo> <comp:element-def name="font" bytecode="16"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:attribute name="id" type="xs:string" use="optional"/> <xs:attribute name="src" type="xs:anyURI" use="required"/> <xs:attribute name="size" type="xs:integer" use="optional"/> <xs:attribute name="fgstatic" type="xs:string" use="optional"/> <xs:attribute name="fgcolor" type="xs:string" use="optional"/> <xs:attribute name="fgfocused" type="xs:string" use="optional"/> <xs:attribute name="fgactivated" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>				

A font element is used to reference font resources and provide font-formatting guidelines.

7.2.1.1 Id attribute

id=CDATA

The id attribute is used to give a unique id to the font definition. The id value can be one of the seven standard WML text emphasis constructs (strong, em, b, i, u, big, small), or can be any arbitrary name (helvetica, heading, etc.).

Choosing an id for a font definition has implications for how that definition can be applied to text in the WTVML markup. A font definition with a standard WML text emphasis id can be applied directly to text in the markup as follows:

```

<card>
  <p>
    <em>formatted using the emphasis definition</em>
    <br/>
    <b>formatted using the bold definition</b>
    <br/>
    <i>formatted using the italic definition</i>
    <br/>
    <strong>formatted using the strong definition</strong>
    <br/>
    <small>formatted using the small definition</small>
    <br/>
    <big>formatted using the big definition</big>
    <br/>
    <u>formatted using the underline definition</u>
  </p>
</card>

```

A font definition that has an arbitrarily named id can only be applied as the value of the font attribute of a high-order display element.

```
<callrate font="Helvetica" ...>
```

7.2.1.2 Src attribute

```
src=href
```

The src attribute specifies the module in which the font resource is contained, along with its pre-allocated resource identifier.

EXAMPLE: `src="http://resources/browser.res#102"`

It is also possible to specify the location of the font resource using relative addressing:

```
src="#101"
```

When relative addressing is used, the Browser should look for the font resource in the default resource directory.

7.2.1.3 Size attribute

```
size="10"
```

The size attribute specifies in pixels the desired vertical size of the font. This attribute is applied to vector fonts only. The Browser should ignore this attribute for bitmap fonts.

7.2.1.4 Fgstatic attribute

```
fgstatic="(#rrggbb|transp)"
```

The fgstatic attribute specifies the foreground colour of a static text element. Static text is text that is not "clickable".

7.2.1.5 Fgcolor attribute

```
fgcolor="(#rrggbb|transp)"
```

The fgcolor attribute specifies the foreground colour of an enabled text element. This should be a colour that distinguishes "clickable" text from static text.

7.2.1.6 Fgfocused attribute

```
fgfocused=CDATA
```

The fgfocused attribute determines the foreground colour of an enabled text element when it has focus.

7.2.1.7 Fgactivated attribute

```
fgactivated=CDATA
```

The fgactivated attribute determines the foreground colour of an enabled text element when it is active.

7.2.1.8 Example

```
<font id="strong" src="http://resources/browser.res#102" size="0" fgstatic="#EBEBEB" fgcolor="#EBEBEB" fgfocused="#f0f0f0" fgactivated="#f0f0f0"/>
```

7.2.2 Fontset element

diagram					
namespace	http://waptv.com/xsd/fontset				
children	fontset:font				
attributes	Name	Type	Use	Default	Fixed
	id	xs:ID	optional		
source	<pre><xs:element name="fontset"> <xs:annotation> <xs:documentation/> <xs:appinfo> <comp:element-def name="fontset" bytecode="15" isroot="true"/> <comp:allowed-attribute name="xmlns" discard="true"/> <comp:allowed-attribute name="xmlns:comp" discard="true"/> <comp:allowed-attribute name="xmlns:wtvml" discard="true"/> <comp:allowed-attribute name="xmlns:xsi" discard="true"/> <comp:allowed-attribute name="xsi:schemaLocation" discard="true"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="fontset:font" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute name="id" type="xs:ID" use="optional"/> </xs:complexType> </xs:element></pre>				

The fontset element is the parent element used to contain one or more font definitions in a fontset XML file.

The fontset is defined in a separate XML document referenced from the fontsrc attribute of the modeset XML file.

In most cases it is desirable to include a default font definition in any fontset. This is the font definition that will be applied to all text that is not explicitly associated with a font definition.

The default font is defined in the same way as any other font, but the id attribute is left with a value of null.

7.2.2.1 Example

```
<?xml version="1.0">
<DOCTYPE fontset PUBLIC "-//SKYINTERACTIVE//DTD fontset 1.0//EN"
http://waptv.com/dtd/modeset_1.0.dtd>

<fontset>

<!-- this is the default font definition -->

  <font id="" src="http://skyenq/skyenq.res#1000" fgstatic="#EBEBEB" size="0" fgcolor="#EBEBEB"
fgfocused="#fe9901" fgactivated="#f0f0f0"/>

<!-- end default font definition -->

  <font id="strong" src="http://resources/browser.res#102" size="0" fgstatic="#EBEBEB"
fgcolor="#EBEBEB" fgfocused="#f0f0f0" fgactivated="#f0f0f0"/>
  <font id="small" src="#100" fgstatic="#EBEBEB" size="0" fgcolor="#EBEBEB" fgfocused="#f0f0f0"
fgactivated="#f0f0f0"/>

  <font id="big" src="#102" fgstatic="#EBEBEB" size="0" fgcolor="#EBEBEB" fgfocused="#fe9901"
fgactivated="#fe9901"/>
  <font id="b" src="#101" fgstatic="#EBEBEB" size="0" fgcolor="#EBEBEB" fgfocused="#EBEBEB"
fgactivated="#EBEBEB"/>
  <font id="u" src="#100" fgstatic="#fe9901" size="0" fgcolor="#fe9901" fgfocused="#f0f0f0"
fgactivated="#f0f0f0"/>

  <font id="em" src="http://domain/filename.res#1000" fgstatic="#000099" size="0"
fgcolor="#000099" fgfocused="#f0f0f0" fgactivated="#f0f0f0"/>
  <font id="i" src="#100" fgstatic="#b6a809" size="0" fgcolor="#b6a809" fgfocused="#f0f0f0"
fgactivated="#f0f0f0"/>
```

```

<font id="menu"      src="#100" fgstatic="#07084f" size="0" fgcolor="#EBEBEB" fgfocused="#fe9901"
fgactivated="#fe9901"/>
<font id="callrate" src="#100" fgstatic="#EBEBEB" size="0" fgcolor="#EBEBEB" fgfocused="#f0f0f0"
fgactivated="#f0f0f0"/>
<font id="title"    src="#102" fgstatic="#ebbe0f" size="0" fgcolor="#ebbe0f" fgfocused="#f0f0f0"
fgactivated="#f0f0f0"/>
<font id="hint"     src="#100" fgstatic="#EBEBEB" size="0" fgcolor="#EBEBEB" fgfocused="#f0f0f0"
fgactivated="#f0f0f0"/>

</fontset>

```

8 WTVML 6.3 Doset specification

8.1 Wtvml:do elements

Within the WTVML 6.3 environment, wtvml:do elements are referenced according to the following hierarchy:

- Doset;
- Modeset;
- Template;
- Card.

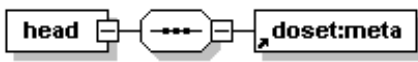
A lower level do overrides a higher level do of the same name. On any card, the active menu dos consist of the do elements specified in the card, plus any other dos specified in the higher level constructs.

The hierarchic model allows easy selective replacement and remapping of menu do items as required without having to redefine the entire menu.

8.2 Cache control

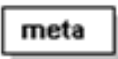
Cache behaviour for doset files can be controlled through the meta element in the head of the dosetinstate.xml file, in the same way that cache behaviour for wtvml files can be defined in the meta element.

8.2.1 Head element

diagram	
namespace	http://waptv.com/xsd/doset
children	doset:meta
used by	element doset
source	<pre> <xs:element name="head"> <xs:complexType> <xs:sequence> <xs:element ref="doset:meta"/> </xs:sequence> </xs:complexType> </xs:element> </pre>

The head element can contain a single meta element, and has no attributes.

8.2.2 Meta element

diagram						
namespace	http://waptv.com/xsd/doset					
used by	element head					
attributes	Name	Type	Use	Default	Fixed	Annotation
	http-equiv	xs:string	optional			
	content	xs:string	optional			
source	<pre><xs:element name="meta"> <xs:complexType> <xs:attribute name="http-equiv" type="xs:string" use="optional"/> <xs:attribute name="content" type="xs:string" use="optional"/> </xs:complexType> </xs:element></pre>					

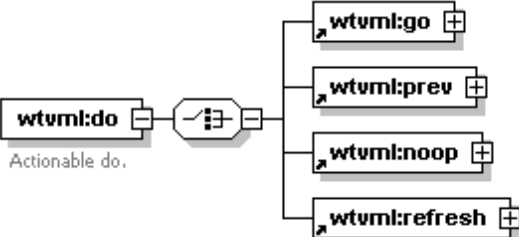
8.2.2.1 Http-equiv attribute

The http-equiv attribute provides a mechanism to set an http header, where the http-equiv value is the header name, and content is the header value. The network's transcoding server will convert an "http-equiv" meta value to an http response header before sending the file to the microbrowser browser.

8.2.2.2 Content attribute

When set to a value of "no-cache", the http-equiv meta value will instruct the microbrowser not to cache the dosetinstance.xml file.

8.2.3 Wtvml:do element

diagram						
namespace	http://waptv.com/xsd/wtvml					
children	wtvml:go wtvml:prev wtvml:noop wtvml:refresh					
used by	elements doset wtvml:card wtvml:fieldset wtvml:p wtvml:pre wtvml:td wtvml:template					
attributes	Name	Type	Use	Default	Fixed	
	type	xs:NMTOKEN	required			
	label	xs:string	optional			
	name	xs:NMTOKEN	optional			
	optional	xs:boolean	optional	false		
	tabindex	xs:NMTOKEN	optional			
	title	xs:string	optional			
	focus	xs:boolean	optional	true		
	id	xs:ID	optional			
	class	xs:string	optional			
	localsrc	xs:string	optional			
	font	xs:string	optional			
	bglocalsrc	xs:string	optional			
	bgoffset	xs:string	optional			
	pos	xs:string	optional			

source	<pre> <xs:element name="do"> <xs:annotation> <xs:documentation>Actionable do. </xs:documentation> <xs:appinfo> <comp:element-def name="do" bytecode="28"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:choice> <xs:element ref="wtvml:go"/> <xs:element ref="wtvml:prev"/> <xs:element ref="wtvml:noop"/> <xs:element ref="wtvml:refresh"/> </xs:choice> <xs:attribute name="type" use="required"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration value="accept"/> <xs:enumeration value="prev"/> <xs:enumeration value="help"/> <xs:enumeration value="reset"/> <xs:enumeration value="options"/> <xs:enumeration value="delete"/> <xs:enumeration value="unknown"/> <xs:enumeration value="vnd.wtv-r"/> <xs:enumeration value="vnd.wtv-g"/> <xs:enumeration value="vnd.wtv-y"/> <xs:enumeration value="vnd.wtv-b"/> <xs:enumeration value="vnd.wtv-sel"/> <xs:enumeration value="vnd.wtv-p"/> <xs:enumeration value="vnd.wtv-m"/> <xs:enumeration value="vnd.wtv-text"/> <xs:enumeration value="vnd.wtv-i"/> <xs:enumeration value="vnd.wtv-0"/> <xs:enumeration value="vnd.wtv-1"/> <xs:enumeration value="vnd.wtv-2"/> <xs:enumeration value="vnd.wtv-3"/> <xs:enumeration value="vnd.wtv-4"/> <xs:enumeration value="vnd.wtv-5"/> <xs:enumeration value="vnd.wtv-6"/> <xs:enumeration value="vnd.wtv-7"/> <xs:enumeration value="vnd.wtv-8"/> <xs:enumeration value="vnd.wtv-9"/> <xs:enumeration value="vnd.wtv-up"/> <xs:enumeration value="vnd.wtv-down"/> <xs:enumeration value="vnd.wtv-left"/> <xs:enumeration value="vnd.wtv-right"/> <xs:enumeration value="vnd.wtv-home"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="label" type="xs:string" use="optional"/> <xs:attribute name="name" type="xs:NMTOKEN" use="optional"/> <xs:attribute name="optional" type="xs:boolean" use="optional" default="false"/> <xs:attribute name="tabindex" type="xs:NMTOKEN" use="optional"/> <xs:attribute name="title" type="xs:string" use="optional"/> <xs:attribute name="focus" type="xs:boolean" use="optional" default="true"/> <xs:attribute name="id" type="xs:ID" use="optional"/> <xs:attribute name="class" type="xs:string" use="optional"/> <xs:attribute name="localsrc" type="xs:string" use="optional"/> <xs:attribute name="font" type="xs:string" use="optional"/> <xs:attribute name="bglocalsrc" type="xs:string" use="optional"/> <xs:attribute name="bgoffset" type="xs:string" use="optional"/> <xs:attribute name="pos" type="xs:string" use="optional"/> </xs:complexType> </xs:element> </pre>
---------------	--

8.2.3.1 Type attribute

The doctype attribute defines which key binding should trigger the do action. See clause 5.2.3.1 for full details of doctype.

8.2.3.2 Label attribute

The label attribute specifies the text content to be displayed in the menu for this do action.

8.2.3.3 Name attribute

The name attribute can be used as a mechanism to customize the sort order of menu do elements. All do elements are sorted in ascending alphabetical order by their name attribute.

8.2.3.4 Tabindex attribute

The tabindex attribute is used to customize a tab order for the menu elements. Tab order can be defined independently of the visible order of menu items.

8.2.3.5 Title attribute

The title attribute can be used to define text content to be displayed in the hint text area of the screen when this do has focus.

8.2.3.6 Localsrc attribute

localsrc=CDATA

The microBrowser should attach a localsource image to most actionable do. The image corresponds to the dotype specified for the do. A do with a dotype of vnd.wtv-g will have the "green" button icon appear onscreen beside the do.

By default, dos of "unknown" dotype will have the "a" icon image drawn on screen beside the do.

Application authors can replace the default image with another localsrc image, including images of their own design, or with a blank image, using the icon attribute.

To remove the default icon from the do, specify icon="blank", or icon="" in the do element.

8.2.4 Doset element

diagram	
namespace	http://waptv.com/xsd/doset
children	wtvml:do
source	<pre><xs:element name="doset"> <xs:annotation> <xs:documentation/> <xs:appinfo> <comp:element-def name="doset" bytecode="17" isroot="true"/> <comp:allowed-attribute name="xmlns" discard="true"/> <comp:allowed-attribute name="xmlns:comp" discard="true"/> <comp:allowed-attribute name="xmlns:wtvml" discard="true"/> <comp:allowed-attribute name="xmlns:xsi" discard="true"/> <comp:allowed-attribute name="xsi:schemaLocation" discard="true"/> </xs:appinfo> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="wtvml:do" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element></pre>

The doset element is a collection of do subelements. A doset is defined in a standalone XML file, and is referred to from the dosrc attribute of the modeset element of a modeset XML file.

Dosets are defined independently of other wtvml markup to enable a do hierarchy that can span an entire portal. Applications can simply reference an external doset to ensure consistency in menu structure across applications.

The doset XML file can have any name, and can be located at any position within the hierarchy of the service's WML site.

Typically, the doset file is called doset.xml and is found in the /doset/ subdirectory of the WML site.

8.2.4.1 Doset example file

```
<?xml version="1.0"?>
<doset>
<do type='vnd.wtv-1' label='Doset1'>
  <noop>
    <setattr name='wtv.alert' value='Doset 1' />
  </noop>
</do>
<do type="vnd.wtv-0" name="Help" label="Help">
<go method="post" href="mercury">
<postfield name="action" value="GoHelp"/>
<postfield name="accum_del" value="" />
<postfield name="ctx" value="false"/>
</go>
</do>
</doset>
```

9 Document Object Model

A WTVML 6.3 microbrowser must incorporate a Document Object Model (DOM) that permits directly setting and retrieving object attribute values within the WTVML platform environment.

All objects are identified by a unique id.

Most of the onscreen objects defined within a modeset.xml file have permanent ids which cannot be modified by the application author. These are as follows.

Object	id	attributes
The WTVML microbrowser application	Wtv	visible version mode mode.fragment locale video audio alert timer flushcache rand launch box xml-debug mem-osd
The http element	http	connectstate title hint connectdelay
The main element	Main	As per browser element in modeset.xsd
The menu element	Menu	As per menu element in modeset.xsd
The dialog element	Dialog	As per dialog element in modeset.xsd
The av element	Av	As per av element in modeset.xsd
The status element	Status	As per status element in modeset.xsd
The callrate element	Callrate	As per callrate element in modeset.xsd
The hint element	Hint	As per hint element in modeset.xsd
The title element	Title	As per title element in modeset.xsd
The table element	Table	As per table element in modeset.xsd
The resume element	resume	As per resume element in modeset.xsd
The scroll object	Scroll	As per scroll element in modeset.xsd

The application author can directly manipulate the attribute values of all these objects using `setattr` statements within their WTVML code. The attributes and usage for all objects except the `wtv` object is described in clause 6 WTVML 6.3 Modeset specification.

9.1 Setting object attribute values

The `setattr` statement can be used to directly set attribute values within most objects in the WTVML environment.

The name attribute identifies the object attribute to be modified, using the syntax:

- `objectid.attribute`.

The value attribute defines the value for that variable.

A very wide variety of attributes can be specified in this way. The `setattr` statement can be used in place of `modeset` changes to manipulate the various attributes of microbrowser objects as required.

9.1.1 Using the '&' character in setattr statements

Standard WML syntax requires certain control characters to be "escaped" in order for them to appear correctly within WML layout.

It is necessary to use the standard WML escape code `&` in place of the ampersand character in order to use `setattr` to set a complex URL that includes the ampersand character.

```
<setattr name="URL" value=http://mysite/mydeck?a=b&amp;c=d/>
```

9.2 Retrieving attribute values

It is possible to retrieve any attribute value using `$` substitution syntax. To programmatically retrieve the current mode, an application author can query the `wtv.mode.fragment` attribute as follows:

- `$(wtv.mode.fragment)`

9.3 Wtv object

9.3.1 Visible attribute

Attribute used to show and hide the WTVML microbrowser application. By setting the attribute to a value of "false", all browser objects should disappear from the TV screen, leaving only full screen video and the resume object (if defined) visible.

- `<setattr name="wtv.visible" value="true"/>`
- `<setattr name="wtv.visible" value="false"/>`

9.3.2 Version attribute

Attribute used to retrieve microbrowser version number.

```
$(wtv.version)
```

9.3.3 Mode attribute

Attribute used to set a new modeset or retrieve the current modeset.

```
<setattr name="wtv.mode" value=http://mysite/mymode.xml/>
```

```
$(wtv.mode)
```

9.3.4 Mode.fragment attribute

Attribute used to set or retrieve an individual mode within a modeset.

```
<setattr name="wtv.mode.fragment" value="#mode1"/>
```

```
$(wtv.mode.fragment)
```

This can also be used to set and retrieve the global translucency value. See clause 6.3.26.3 for details.

```
<setattr name="wtv.mode.translucency" value="128"/>
```

```
$(wtv.mode!translucency)
```

9.3.5 Locale attribute

Attribute used to set or retrieve the current locale. Currently, only two locales are supported, UK (for United Kingdom) and IR (for Republic of Ireland).

```
<setattr name="wtv.locale" value="UK"/>
```

```
$(wtv.locale)
```

9.3.6 Video attribute

Attribute used to retrieve indication whether video is currently playing in the av window or not.

```
$(wtv.video) returns values of "video" or "novideo"
```

9.3.7 Audio attribute

Attribute used to retrieve indication whether audio is currently playing or not.

```
$(wtv.audio) returns values of "audio" or "noaudio"
```

9.3.8 Alert attribute

Attribute used to launch the dialog object with a green OK button to dismiss the dialog. The value of the wtv.alert attribute sets the text for the popup. Positioning and appearance is defined by the dialog element in the mode.

9.3.9 Timer attribute

Attribute used to set or retrieve the value of a timer in a WTVML card.

```
<setattr name="timer.value" value="40" />
```

9.3.10 Flushcache attribute

Attribute used to control the flushing of cached resources.

The example below will force a flush of all currently unused resources.

- `<setattr name="wtv.flushcache" value="true"/>`The example below will force a flush of all currently unused cache contents with a cacheset identifier of "a".
- `<setattr name="wtv.flushcache" value="a"/>`The default cacheset value is "0", so the example below will flush all cache content except those that have had a cacheset identifier explicitly assigned.
- `<setattr name="wtv.flushcache" value="0"/>`

9.3.11 Rand attribute

Attribute used to generate a random integer within predefined limits.

The example below will set the variable wtv.rand to a random integer value between 1 and 20.

```
<setattr name="wtv.rand" value="1,20"/>
```

9.3.12 Launch attribute

Attribute used to set or retrieve the value of WTV-launch. The value of this attribute is presented to the CP webserver in the HTTP request header.

WTV-launch is set at launch of the service by the ISM.

The default value of wtv.launch is null. All service launches from the ISM have WTV-Launch value of null.

Other launchpoints, such as banners or interactive advertising (IA) redbutton launches, will have meaningful values for WTV-launch. It is not necessary for individual IA init tracks to be developed for each time and channel in the advertising schedule as the time and channel identification are available for other sources: time is recorded by default on the webserver for every request, channel is programmatically retrievable through the \$(av.channel).

```
<setattr name="wtv.launch" value="somevalue"/>
```

```
$(wtv.launch)
```

9.3.13 Stb attribute

Attribute used to enable programmatic retrieval of set-top-box details at runtime. Supported attributes are:

Attribute	Description
wtv.stb.opentv-version	Returns OpenTV middleware version number.
wtv.stb.manufacturer	Returns set-top-box manufacturer name.
wtv.stb.manufacturer-version	Returns set-top-box type / model name.
wtv.stb.info	Returns STB hardware id number.
wtv.stb.heap-available	Returns in bytes total size of memory currently available.
wtv.stb.heap-biggest	Returns in bytes size of largest contiguous chunk of memory currently available.

```
$(wtv.stb!opentv-version)
$(wtv.stb!manufacturer)
$(wtv.stb!manufacturer-version)
$(wtv.stb!info)
$(wtv.stb!heap-available)
$(wtv.stb!heap-biggest)
```

9.3.14 Xml-debug attribute

Attribute used to toggle dump of WTVML content XML transcript to application level debug.

This requires using a "live debug" set-top-box running a debug build of OpenTV middleware. This kind of debug box delivers application debug over serial port as the application is running.

Setting the wtv.xml-debug attribute value to "true" will force the browser to dump the WTVML content XML transcript along with browser application debug.

```
<setattr name="wtv.xml-debug" value="true"/>
```

```
<setattr name="wtv.xml-debug" value="false"/>
```

```
$(wtv!xml-debug)
```

9.3.15 Mem-osd attribute

Attribute used to toggle the display of live memory information on the OSD layer of the set-top-box display.

```
<setattr name="wtv!mem-osd" value="true"/>
<setattr name="wtv!mem-osd" value="false"/>
$(wtv!mem-osd)
```

9.4 Http element

The http object can only be accessed via setattr or via variable substitution. It provides a mechanism to control connection status or retrieve current connection status from within the WTVML markup.

9.4.1 Connectstate attribute

The connectstate attribute can be used to set or retrieve the connection status for the browser. When queried it will return one of the following values:

Value	Description
disconnecting	The browser is going offline.
disconnected	The browser is offline.
connected	The browser is online.
connecting	The browser is going online.
active	The browser is connected and an http request is active.

The connectstate attribute can also be used via setattr to force the browser to connect or to disconnect:

```
<setattr name="http.connectstate" value="connect"/>
<setattr name="http.connectstate" value="disconnect"/>
```

9.4.2 Title attribute

Returns the value of a short description for tariff as specified in browser.rc file. This can be used by an application author to retrieve the current tariff information for a service. The tariff cannot be set from within the WTVL code.

```
<wml>
  <card>
    <p>
The browser is currently $(http.connectstate). The phone charge for this service is $(http.title)
    </p>
  </card>
</wml>
```

9.4.3 Hint attribute

Returns the value of a long description for tariff as specified in browser.rc file.

9.4.4 Connectdelay attribute

Used to specify in seconds a duration to expire before a connection is initiated. The following example code would initiate a connection after a 30 second delay.

```
<setattr name="http.connectdelay" value="30"/>
<setattr name="http.connectstate" value="connect"/>
```

The connectdelay function can be used in conjunction with the wtv.rand function. This is useful in interactive advertising scenarios where it is desirable to manage many simultaneous connection attempts to provide optimum user experience. The following code will randomly spread the connection load across a 30-second time window.

```
<setattr name="wtv.rand" value="0,30"/>
<setattr name="http.connectdelay" value="$(wtv.rand)"/>
```



```
<setattr name="http.connectstate" value="connect"/>
```

See clause 9.3.11 for details on wtv.rand.

9.4.5 Tpsec attribute

Used to trigger establishment of secure connection.

```
<setattr name="http.tpsec" value="true"/>
```

9.4.6 Example 1

It is possible to use setattr to trigger a mode change from any WML event. In this example, the event type is "onactivate". The action is to change to the mode with the id "mode2" within the current modeset.

This could be triggered by navigating focus to, then activating a WML table cell containing this action.

```
<onevent type='onactivate'>
  <noop>
    <setattr name='wtv.mode' value='#mode2' />
  </noop>
</onevent>
```

9.4.7 Example 2

It is possible to use setattr statements to swap images. In this example, the event type is "onenterforward". Using the noop action to perform an image change has the advantage that the image can be changed without requiring the browser to lay out any new WML. This technique delivers very efficient image manipulation functionality.

```
<onevent type="onenterforward" >
  <noop >
    <setattr name="banner.src" value="http://mysite/banner1.jpg" />
  </noop>
</onevent>
```

9.5 URL syntax

A variety of URL structures are supported in WTVML. These are:

URL structure	Description
http:	Standard http.
https:	Secure http : specifies using secure protocols.
exit:	Quit browser application.
return:	Return to launch-point home card. This is launch point specific, and will function to return the user to an some other appropriate location as specified by the launch point.
blank:	Clears down browser of current WML content by loading a blank WML card.

9.5.1 Browser services

The regular URL syntax links to various browser services using the standard dialup modem connection to retrieve content from a webserver.

```
http://<domain>/...
```

Secure http communications can be specified in exactly the same way it is specified on the web.

```
https://<domain>/...
```

9.5.2 Blank: URL

The blank: URL that can be used to delete the contents of any browser other than the main browser.

It can be used to save memory in a similar way to loading an empty deck into a browser, but this method avoids the requirement to have a blank.wml deck, and is far more efficient because the browser is completely empty and does not contain even a blank.wml.

When the blank: URL is loaded into a browser pane, the current deck's gadgets are deleted and its associated XML structures, bytecode and module become candidates for flush from memory.

The blank: URL can be used in any browser but the main browser as the main browser.

It can be used in either a `<go>` or a `<setattr>` as follows: `<go href="blank:" target="popup" />`
`<go href="blank:" />`
`<setattr name="popup.href" value="blank:" /`

10 WML script

10.1 Introduction

WML Script is an extended subset of the ECMA Script scripting language, which in turn is derived from the JavaScript™ scripting language.

10.2 Base WML Script specification

See WAP Forum WML Script Specification at <http://wapforum.org>.

10.3 WMLScript for the WTVML 6.3 microbrowser

Certain modifications have been made to WMLScript for the WTVML 6.3 microbrowser. These changes are detailed in this clause.

Service authors are advised to refer to the WAP Forum WML Script Specification documentation for full documentation of the WML Script language.

The WMLBrowser library contains functions by which WMLScript can access the associated WML context. Some of the original WAP Forum WMLBrowser library functions have been modified to address the multiple browser capability of the WTVML standard.

10.3.1 GetAttr function

The `getAttr` function returns the value of the attribute with the given *name* in the given browser context. If no context is given, object ID is implied. Returns an empty string if the given attribute does not exist. Attribute name follows the WTVML microbrowser syntax for attributes.

FUNCTION:	<code>getAttr(name)</code> .
FUNCTION ID:	7.
PARAMETERS:	<i>name</i> = String.
RETURN VALUE:	String or invalid.
EXCEPTIONS:	If the syntax of the attribute name is incorrect an <code>invalid</code> value is returned.

10.3.1.1 Example

```
var a = WMLBrowser.getAttr("main!href");
```

10.3.2 SetAttr function

Returns true if the attribute with the given *name* is successfully set to contain the given *value* in the given browser context, false otherwise.

If no context is given, object ID is implied. Attribute name must follow the microbrowser syntax for attributes. Attribute value must be legal XML CDATA.

FUNCTION:	setAttr(name, value).
FUNCTION ID:	8.
PARAMETERS:	name = String, value = String.
RETURN VALUE:	Boolean or invalid.
EXCEPTIONS:	If the syntax of the attribute name or its value is incorrect an <i>invalid</i> value is returned.

10.3.2.1 Example

```
var a = WMLBrowser.setAttr("main!href", "http://www.waptv.com" )
```

10.3.3 GoTarget function

Loads nominated href into target browser. The http-equiv parameter is equivalent to the "cache-control" attribute in a go.

FUNCTION:	goTarget(href, target, http-equiv).
FUNCTION ID:	9.
PARAMETERS:	href = String, target=String, http-equiv=String.

10.3.3.1 Example

```
WMLBrowser.goTarget("http://test.waptv.co.uk/target_go.wml", "second", "no-cache")
```

10.3.4 GetVar function

The regular WML Script getVar function has been extended to support cross-browser variable references of the form browser.variable.

FUNCTION:	getVar(<i>name</i>).
FUNCTION ID:	0.
PARAMETERS	<i>name</i> = String.

10.3.4.1 Example

```
WMLBrowser.getVar("main.var1")
```

10.3.5 SetVar function

The regular WML Script setVar function has been extended to support cross-browser variable references of the form browser.variable.

FUNCTION:	setVar(name,value).
FUNCTION ID:	1.
PARAMETERS:	name=String, value=String.

10.3.5.1 Example

```
WMLBrowser.setvar("main.var1", "somevalue")
```

11 Validating markup

Application authors can use third party XML validating tools to check their wtvml markup, modeset, fontset and doset files against published WTVML definitions, and eliminate any syntax or nesting errors that may exist in their code.

Although successful validation does not necessarily mean that an instance of markup will work as expected in the microbrowser, it does mean that most common errors caused by compilation failure, malformed markup and simple nesting and notation errors can be identified and eliminated.

11.1 Validating wtvml files

WTVML files pose a special problem for validation, because the underlying WAP Forum WML specification requires escape codes for special characters such as "&", "<", ">", and the XSD notation does not natively support the entity references that are required to validate these codes.

To work around this problem, application authors who wish to validate their code against the wtvml XSD definition are advised to include an internal DTD subset within any WTVML instance files that include escaped characters. The internal subset should be inserted at the top of the instance file to be validated.

It is not compulsory to include the internal DTD subset, as it is used for validation only. WTVML files without the subset should still be successfully transformed by the network edge server and displayed in the WTVML 6.3 microbrowser.

11.1.1 Internal DTD subset

The internal DTD subset provides a mechanism within the WTVML instance document to allow correct entity resolution of escape codes for special characters. This in turn allows XML validating tools to successfully validate instance files that contain any of these entity references.

An example of the internal subset code is included below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WML [
<!ENTITY nbsp "&#160;">
<!ENTITY quot "&#34;">
<!ENTITY amp "&#38;#38;">
<!ENTITY apos "&#39;">
<!ENTITY lt "&#38;#60;">
<!ENTITY gt "&#62;">
<!ENTITY shy "&#173;">
]>
<wml>
...
</wml>
```

11.1.2 Root element

XML validating processors identify the XSD definition to be used for validating any instance file by reading some special "xmlns" attributes added to the root element of the instance file.

These attributes define the namespace of the document, and also point to the URL where a validating XML processor can find the XSD definition against which to validate the file.

The WTVML fragment below includes the required "xmlns" attributes in the wml root element:

```
<wml mode="modeset.xml" xmlns="http://waptv.com/xsd/wtvml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://waptv.com/xsd/wtvml
http://waptv.com/xsd/wtvml_5.0.xsd">
  <card mode="mymode">
    ...
  </card>
</wml>
```

11.2 Validating modeset files

Modeset files pose a special problem for validating XML processors due to the existence of two different do elements in modeset files.

One kind of do belongs to the modeset definition, and describes the appearance of the actionable do elements that will exist in the menu. The other kind is the same as any regular do in WTVML, and represents and actionable element.

The XSD definitions published for the version 5 microbrowser accommodate these conflicting do definitions by using XML namespace notation, which allows the "menu do" to be validated against the modeset.xsd definition, while the "menu do" is validated against the wtvml.xsd definition.

11.2.1 Root element

XML validating processors identify the XSD definition to be used for validating any instance file by reading some special "xmlns" attributes added to the root element of the instance file.

These attributes define the namespace of the document, and also point to the URL where a validating XML processor can find the XSD definition against which to validate the file.

The modeset.xsd definition also refers to the wtvml.xsd definition to enable validation of some instances of the do element.

The modeset example below includes the required "xmlns" attributes in the document headers:

```
<?xml version="1.0" encoding="UTF-8"?>
<modeset fontsrc="fontset.wml" dosrc="doset.xml"
xmlns="http://waptv.com/xsd/modeset"
xmlns:wtvml="http://waptv.com/xsd/wtvml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://waptv.com/xsd/modeset
http://waptv.com/xsd/modeset_5.0.xsd">
  <mode>
    ...
  </mode>
</modeset>
```

The xmlns attributes in the root element are required to tell the validator what namespace this instance document belongs to, and the location of the associated XSD definition.

11.2.2 Do elements

To enable validation of do elements in modeset files, application authors should use the "wtvml:" namespace prefix for the "menu do" instance of the do element, and any child elements that may also exist.

Although it is recommended best practice, it is not compulsory to use this namespaced notation for modeset files. It is useful for validation only. Modeset files without namespaced do elements should still be successfully compiled and displayed in a WTVML microbrowser. This is to ensure backward compatibility with pre-existing modeset instances that do not include the namespaced element notation.

```
<?xml version="1.0" encoding="UTF-8"?>
<modeset fontsrc="fontset.xml" dosrc="doset.xml" localsrc="" xmlns="http://waptv.com/xsd/modeset"
xmlns:wtvml="http://waptv.com/xsd/wtvml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://waptv.com/xsd/modeset
http://waptv.com/xsd/modeset_5.0.xsd">
  <mode id="sample_mode" bgimage="sample_bg.mpg">
    <browser rect="20,70,200,200"/>
    <!-- -->
    <!-- The do element can only appears as a child -->
    <!-- element of menu, and defines the *appearance* -->
    <!-- of do elements within the menu object -->
    <!-- -->
    <menu rect="insert menu rect coordinates here">
      <do font="insert font id here"/>
    </menu>
    <!-- -->
    <!-- The wtvml:do element exists loose in a -->
    <!-- mode definition, and can only be validated -->
    <!-- correctly if specified under the 'wtvml' -->
```

```

<!-- namespace. This is to distinguish it from the -->
<!-- modeset:do element which exists in the -->
<!-- 'modeset' namespace -->
<!-- -->
<wtvml:do id="insert_id_here" type="unknown" label="insert label here">
  <wtvml:go href="insert href here"/>
</wtvml:do>
</mode>
</modeset>

```

11.3 Validating fontset files

XML validating processors identify the XSD definition to be used for validating any instance file by reading some special "xmlns" attributes added to the root element of the instance file.

These attributes define the namespace of the document, and also point to the URL where a validating XML processor can find the XSD definition against which to validate the file.

The fontset example below includes the required "xmlns" attributes in the document headers:

```

<?xml version="1.0" encoding="UTF-8"?>
<fontset xmlns="http://waptv.com/xsd/fontset"
xmlns:wtvml="http://waptv.com/xsd/wtvml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://waptv.com/xsd/fontset
http://waptv.com/xsd/fontset_5.0.xsd">
  <font/>
  ...
</fontset>

```

11.4 Validating doset files

Dosets provide a high level construct for "menu do" elements to be defined, from where they can be easily and uniformly applied across multiple WTVML decks and cards.

Because the do elements that can be defined in a doset are in fact exactly the same kind of do that can be defined in a WTVML file, they are also validated against the original wtvml definition.

In the same way that recommended best practice for "menu dos" is that they should be specified using the wtvml: prefix in modesets, so too should the wtvml: namespace prefix be used for dos in dosets.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- The 'xmlns' attributes identify the namespaces and -->
<!-- locations of XSD definitions to be used by XML validating -->
<!-- tools to validate this file -->
<!-- -->
<doset xmlns="http://waptv.com/xsd/doset" xmlns:wtvml="http://waptv.com/xsd/wtvml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://waptv.com/xsd/doset
http://waptv.com/xsd/doset_5.0.xsd">
  <!-- -->
  <!-- Do elements in a doset are specified under the wtvml -->
  <!-- namespace as they are validated under the wtvml XSD -->
  <!-- -->
  <wtvml:do type="unknown" label="insert label here" title="insert hint text content here">
    <wtvml:go href="insert href here"/>
  </wtvml:do>
</doset>

```

11.4.1 Root element

XML validating processors identify the XSD definition to be used for validating any instance file by reading some special "xmlns" attributes added to the root element of the instance file.

These attributes define the namespace of the document, and also point to the URL where a validating XML processor can find the XSD definition against which to validate the file.

The doset example below includes the required "xmlns" attributes in the document headers:

```
<?xml version="1.0" encoding="UTF-8"?>  
<doset xmlns="http://waptv.com/xsd/doset"  
xmlns:wtml="http://waptv.com/xsd/wtml"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://waptv.com/xsd/doset/http://waptv.com/xsd/doset_5.0.xsd">
```

Annex A (informative): Bibliography

ECMA-262 (1999): "ECMAScript Language Specification".

JavaScript™ (1997): "The Definitive Guide", David Flanagan.

IETF RFC 1738 (1994): "Uniform Resource Locators (URL)".

IETF RFC 1808 (1995): "Relative Uniform Resource Locators".

IETF RFC 2045 (1996): "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

IETF RFC 2048 (1996): "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures".

IETF RFC 2616 (1999): "Hypertext Transfer Protocol -- HTTP/1.1".

IETF RFC 2119 (1997): "Key words for use in RFCs to Indicate Requirement Levels".

"The Unicode Standard: Version 2.0" (1996) The Unicode Consortium, Addison-Wesley Developers Press.

NOTE: See <http://www.unicode.org/>.

"Wireless Application Protocol Architecture Specification" (1998) WAP Forum.

"Wireless Markup Language Specification" (1999) WAP Forum.

"WMLScript Language Specification version 1.1" (1999) WAP Forum.

"WMLScript Standard Libraries Specification version 1.1" (1999) WAP Forum.

NOTE: See <http://www.wapforum.org/>

"Extensible Markup Language (XML) 1.0 W3C Recommendation (1998) REC-xml-19980210".

NOTE: See <http://www.w3.org/TR/REC-xml>.

History

Document history		
V1.1.1	May 2004	Publication