# **UNITED STATES PATENT AND TRADEMARK OFFICE**

Applicant(s): Rempell et al.

National Phase Application of PCT/US2009/039695, Filed April 6, 2009

Title: SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES

Docket No.: XPR.002US0

# **PRELIMINARY AMENDMENT**

Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450

Dear Commissioner:

This correspondence is a Preliminary Amendment for the application filed herewith.

A listing of the claims with any amendments to the claims begins on a new page

immediately after these introductory remarks.

The *Remarks* begin on a new page immediately after such listing of the claims.

# Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of the claims in the application:

# Listing of Claims:

Claim 1 (Currently Amended): 1. A system for generating code to provide content on a display of a <u>platform\_device</u>, said system comprising:

a database of web services obtainable over a network; an authoring tool configured to:

define an object for presentation on the display,

select a component of a web service included in said database,

associate said object with said selected component, and

produce <u>device-specific</u> code that, when executed on the-<u>platform</u>

<u>device</u>, provides said selected component on the display of the <u>platform</u> <u>device</u>.

Claim 2 (Original): The system of Claim 1, where said database includes definitions of input and/or output related to said web service.

Claim 3 (Original): The system of Claim 1, where said component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

Claim 4 (Original): The system of Claim 1, where said object is an input field for a chat.

Claim 5 (Original): The system of Claim 1, where said object is an input field for a web service.

Claim 6 (Original): The system of Claim 1, where said object is an input field usable to obtain said component, where said input field includes a text field, a scrolling text box, a check box, a drop down-menu, a list menu, or a submit button.

Claim 7 (Original): The system of Claim 1, where said component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.

### Preliminary Amendment for National Phase Application of PCT/US2009/039695

Claim 8 (Currently Amended): The system of Claim 1, where said authoring tool is further configured to:

define a phone field or list; and

generate code that, when executed on the <u>platform device</u>, allows a user to supply a phone number to said phone field or list.

Claim 9 (Currently Amended): The system of Claim 1, where said authoring tool is further configured to:

define a SMS field or list; and

generate code that, when executed on the <u>platform device</u>, allows a user to supply an SMS address to said SMS field or list.

Claim 10 (Currently Amended): The system of Claim 1, where said code includes two or more codes, where one of said two or more codes is <u>platform device</u> specific, and where one of said two or more codes in <u>platform device</u> independent.

Claim 11 (Original): The system of Claim 1, where said code is provided over said network.

Claim 12 (Currently Amended): A method for displaying content on a platform <u>device</u> utilizing a database of web services obtainable over a network, said method comprising:

defining an object for presentation on the display;

selecting a component of a web service included in said database;

associating said object with said selected component; and

producing device-dependent code that, when executed on the-platform

device, provides said selected component on the display of the platform device.

Claim 13 (Original): The method of Claim 12, where said database includes definitions of input and/or output related to said web service.

Claim 14 (Original): The method of Claim 12, where said component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

Claim 15 (Original): The method of Claim 12, where said object is an input field for a chat.

Claim 16 (Original): The method of Claim 12, where said object is an input field for a web service.

### Preliminary Amendment for National Phase Application of PCT/US2009/039695

Claim 17 (Original): The method of Claim 12, where said object is an input field usable to obtain said component, where said input field includes a text field, a scrolling text box, a check box, a drop down-menu, a list menu, or a submit button.

Claim 18 (Original): The method of Claim 12, where said component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.

Claim 19 (Currently Amended): The method of Claim 12, further comprising: defining a phone field or list; and

generating code that, when executed on the <u>platform device</u>, allows a user to supply a phone number to said phone field or list.

Claim 20 (Currently Amended): The method of Claim 12, further comprising: defining a SMS field or list; and

generating code that, when executed on the <u>platform device</u>, allows a user to supply an SMS address to said SMS field or list.

Claim 21 (Currently Amended): The method of Claim 12, where said code includes two or more codes, where one of said two or more codes is <u>platform device</u> specific, and where one of said two or more codes in <u>platform device</u> independent.

Claim 22 (Original): The method of Claim 12, further comprising:

providing said code over said network.

Claim 23 (Currently Amended): A method for providing information to a platform <u>device</u> on a network, said method comprising:

accepting a first code over the network, where said first code is platform <u>device</u>-dependent;

providing a second code over the network, where said second code is platform\_device-independent; and

executing said first code and said second code on the <u>platform device</u> to provide web components obtained over the network.

Claim 24 (Original): The method of Claim 23, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

# Preliminary Amendment for National Phase Application of PCT/US2009/039695

Claim 25 (Original): The method of Claim 23, where said component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.

Claim 26 (Original): The method of Claim 23, where said first code and said second code are generated using an authoring tool.

Claim 27 (Original): The method of Claim 23, where said first code is a Player. Claim 28 (Original): The method of Claim 23, where said second code is an Application.

# Remarks

This preliminary amendment is being submitted with the filing of the abovereferenced application in the United States. This Preliminary Amendment amends Claims 1, 8-10, 12, 19-21, and 23. **Claims 1-28 are pending for consideration**.

Applicants respectfully submit that, as amended, all pending claims in this application are patentably distinct over the prior art of record. Reconsideration and allowance of all pending claims in the application are respectfully solicited.

# Amendment to the Claims

Claims 1, 8-10, 12, 19-21, and 23 are amended herewith. No new matter is introduced by these amendments, support for which is provided below.

Claims 1, 8-10, 12, 19-21, and 23 are amended to replace "platform" with "device," and are being made to more clearly recite the subject matter of the invention. As described in the specification, the code runs on a device, which may form part of a platform. See for example, paragraph [0039] of the present Specification.

Applicant respectfully submits that the application is in condition for allowance and action to that end is respectfully solicited. If the Examiner should feel that a telephone interview would be productive in resolving any issues in the case, please telephone the undersigned at the number listed below.

Respectfully submitted,

September 29, 2010

1563 Solano Ave., #206 Berkeley, CA 94705 Tel: (510) 841-4711; Fax: (510) 217-6844 /Steven R. Vosen/ Steven R. Vosen Registration No. 45,186

#### DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence/post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

#### SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES

The specification of which is attached hereto unless the following box is checked:

(X) was filed on <u>April 6, 2009</u> as an International Application Serial Number: <u>PCT/US2009/039695</u> and was amended on September 30, 2010.

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

#### Foreign Application(s) and/or Claim of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDE	R 35 U.S.C. 119
			YES: NO:	
			YES: NO:	

#### **Provisional Application**

I hereby claim the benefit under Title 35, United States Code Section 119(c) of any United States provisional application(s) listed below:

APPLICATION SERIAL NUMBER	FILING DATE
61/166,651	April 3, 2009
61/113,471	November 11, 2008
61/123,438	April 7, 2008

#### U.S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NUMBER	FILING DATE	STATUS(patented/pending/abandoned)

#### POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) listed below to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

#### Steven Vosen, Reg. No. 45186

Send Correspondence to:	Direct Telephone Calls To:
Steven Vosen	Steven Vosen
1563 Solano Ave., #206	510-841-4711
Berkeley, CA 94707	

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are putilishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: Steven Rempell	Citizenship: US
Residence: 38 Washington Street, Novato, CA USA 94947	
Post Office A dress: Same 17	
In Rippin	9/20/2010
Inventor's Signature	Date

DECLARATION	AND POWER	OF ATTORNEY
FOR PATENT A	PPLICATION	(continued)

Full Name of Inventor: <u>David Chrobak</u>		Citizenship: US	
Residence: <u>11 Mt. Tamalpais Drive, Clayton, CA 94517 USA</u>			
Post Office Address: Same	*****		
Inventor's Signature	Date		
Full Name of Inventor: Ken Brown		Citizenship: US	
Residence: 2485 Church Avenue, San Martin, CA 95046 USA			
Post Office Address: Same			
Inventor's Signature	Date		

### DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence/post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

### SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES

The specification of which is attached hereto unless the following box is checked:

(X) was filed on <u>April 6, 2009</u> as an International Application Serial Number: <u>PCT/US2009/039695</u> and was amended on <u>September 30, 2010</u>.

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

#### Foreign Application(s) and/or Claim of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDER 35 U.S.C. 119
			YES: NO:
			YES: NO:

#### **Provisional Application**

I hereby claim the benefit under Title 35, United States Code Section 119(e) of any United States provisional application(s) listed below:

APPLICATION SERIAL NUMBER	FILING DATE
61/166,651	April 3, 2009
61/113,471	November 11, 2008
61/123,438	April 7, 2008

#### U.S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NUMBER	FILING DATE	STATUS(patented/pending/abandoned)

#### POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) listed below to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

#### Steven Vosen, Reg. No. 45186

Send Correspondence to:	Direct Telephone Calls To:
Steven Vosen	Steven Vosen
1563 Solano Ave., #206	510-841-4711
Berkeley, CA 94707	
I hereby declare that all statements made herein of my own knowledge are true and that all state	ments made on information and belief are believed to be true; and further tha

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: Steven Rempell

Citizenship:	$\mathbf{US}$

Residence: <u>38 Washington S</u>	<u>street, Novato,</u>	CA	USA	<u>9494'</u>
-----------------------------------	------------------------	----	-----	--------------

Post Office Address: <u>Same</u>

Inventor's Signature

Date

### DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION (continued)

ATTORNEY DOCKET NO. XPR.002US0

Full Name of Inventor: <u>David Chrobak</u>	Citizenship: <u>US</u>
Residence: 11 Mt. Tamalnais Difer. Clayton, CA 94517 USA	
Post Office Address: Same	
	9/2010
Inventor's Signature	Date
Full Name of Inventor: Ken Brown	Citizenship: <u>US</u>
Residence: 2485 Church Avenue, San Martin, CA 95046 USA	
Post Office Address: <u>Same</u>	
Inventor's Signature	Baie

### DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence/post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

### SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES

The specification of which is attached hereto unless the following box is checked:

(X) was filed on <u>April 6, 2009</u> as an International Application Serial Number: <u>PCT/US2009/039695</u> and was amended on <u>September 30, 2010</u>.

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

#### Foreign Application(s) and/or Claim of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDER 35 U.S.C. 119
			YES: NO:
			YES: NO:

#### **Provisional Application**

I hereby claim the benefit under Title 35, United States Code Section 119(e) of any United States provisional application(s) listed below:

APPLICATION SERIAL NUMBER	FILING DATE
61/166,651	April 3, 2009
61/113,471	November 11, 2008
61/123,438	April 7, 2008

#### U.S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NUMBER	FILING DATE	STATUS(patented/pending/abandoned)

#### POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) listed below to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

#### Steven Vosen, Reg. No. 45186

Send Correspondence to:	Direct Telephone Calls To:
Steven Vosen	Steven Vosen
1563 Solano Ave., #206	510-841-4711
Berkeley, CA 94707	
I hereby declare that all statements made herein of my own knowledge ar	e true and that all statements made on information and belief are believed to be true; and further tha

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: Steven Rempell

Citizenshi	n:	US
C. I CHEROMAN COMMEND	÷.	

<b>Residence:</b>	38	Wash	ington	Street.	Novate.	CA	USA	9494

Post Office Address: Same

Inventor's Signature

Date

# DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION (continued)

ATTORNEY DOCKET NO. XPR.002US0

Full Name of Inventor: David Chrobak	Citizenship: <u>US</u>					
Residence: 11 Mt. Tamalpais Drive, Clayton, CA 94517 USA						
Post Office Address: Same						
Inventor's Signature	Date					
Full Name of Inventor: Ken Brown	Citizenship: <u>US</u>					
Residence: 2485 Church Avenue, San Martin, CA 95046 USA						
Post Office Address: Same						
Kenth pm	2010.09,30					

Inventor's Signature

Date

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	XPR.002US0				
		Application Number					
Title of Invention	SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES						
The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.							

This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

# Secrecy Order 37 CFR 5.2

Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

# **Applicant Information:**

Applic	:ant	1										Remove		
Applic	:ant	Authority 🖲	Inventor	OLe	egal	Representativ	e und	er 35	U.S.C. 11	7	OParty of In	terest under 35 U.S.	C. 118	
Prefix	Gi	ven Name	·			Middle Nar	ne			Fam	ily Name		Suffix	
	St	even				H.				Rem	npell			
Residence Information (Select One) (					$\odot$	US Residenc	y (	<u> </u>	on US Res	sidenc	y 🔿 Active	e US Military Service		
City	y Novato S				St	ate/Province	• (	CA	Countr	y of F	Residence <sup>i</sup>	US		
Citizer	nsh	ip under 37 C	FR 1.41(b	<b>)</b> i	U	S								
Mailing Address of Applicant:														
Addre	SS 1		38 Wash	ington	n Stre	eet								
Addre	ss 2	2												
City		Novato						Stat	te/Provin	ice	CA			
Postal	l Co	de	94947				Οοι	untry <sup>i</sup>	US					
Applic	ant	2										Remove		
Applicant Authority Inventor Clegal Representative under 35 U.S.C. 117 Party of Interest under 35 U.S.C. 1								C. 118						
Prefix	Gi	Given Name				Middle Name			Family Name			Suffix		
	D	avid				Chr			hrobak					
Resid	lenc	e Informatio	n (Select (	One)	$\odot$	US Residency O Non US Reside				sidenc	idency O Active US Military Service			
City	С	ayton			St	ate/Province	. (	CA	Countr	y of F	Residence i US			
Citizer	nshi	ip under 37 C	FR 1.41(b	<b>)</b> i	US	8								
Mailin	g A	ddress of Ap	plicant:											
Addre	SS 1		11 Mt. Ta	amalpa	ais C	Xt.								
Addre	ss 2	2												
City		Clayton						Stat	te/Provin	ice	CA			
Postal	l Co	de	94517				Cou	untry <sup>i</sup>	US					
Applic	:ant	3										Remove		
Applic	:ant	Authority •	Inventor	OLe	egal	gal Representative under 35 U.S.C. 11				117 OParty of Interest under 35 U.S.C. 11			C. 118	
Prefix	Gi	ven Name	ŀ			Middle Name				Family Name			Suffix	
	K	en								Brown				
Resid	lenc	e Informatio	n (Select (	One)	ullet	US Residenc	у (		on US Res	Residency O Active US Military Service				
City	Sa	an Martin			St	ate/Province	. (	CA	Countr	y of F	Residence <sup>i</sup>	US		

### IPR2021-01146 Page 00013

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid UMB control number.												
Application Data Sheet 37 CER 1 76				1 76	Attorney Docket Number			XPR.00	2US0			
Application Data Sheet 37 CI K 1.70			1.70	Application Number								
Title of Invention SYSTEMS AND METHODS FOR PR					FOR PRESEN	TING INF	ORMATI	ON ON N	10BILE DE	EVICES		
Citizenshi	p under	37 C	FR 1.41(b) <sup>i</sup>	US								
Mailing A	ddress o	of Ap	plicant:									
Address 1			2485 Church A	venue								
Address 2	2											
City	San Ma	artin				State/Province			CA			
Postal Code 95046			Co	ountry <sup>i</sup>	US							
All Inventors Must Be Listed - Additional Inventor Information blocks may be Add Add												

# **Correspondence Information:**

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).					
An Address is being provided for the correspondence Information of this application.					
Customer Number 40280					
Email Address     svosen@PhDpatents.com     Add Email     Remove Email					

# **Application Information:**

Title of the Invention	SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES			
Attorney Docket Number	XPR.002US0     Small Entity Status Claimed			
Application Type	Nonprovisional			
Subject Matter	Utility			
Suggested Class (if any)		Sub Class (if any)		
Suggested Technology C	enter (if any)			
Total Number of Drawing	g Sheets (if any) Suggested Figure for Publication (if any)			
Dublication Inform				

### Publication Information:

Request Early Publication (Fee required at time of Request 37 CFR 1.219)
<b>Request Not to Publish.</b> I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application has not and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

# **Representative Information:**

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Enter either Customer Number or complete the Representative Name section below. If both sections are completed the Customer Number will be used for the Representative Information during processing.

Please Select One:	Customer Number	O US Patent Practitioner	<ul> <li>Limited Recognition (37 CFR 11.9)</li> </ul>
--------------------	-----------------	--------------------------	---

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	XPR.002US0
		Application Number	
Title of Invention SYSTEMS AND METHODS I		FOR PRESENTING INFORMAT	TON ON MOBILE DEVICES
Customer Number	40280		

# **Domestic Benefit/National Stage Information:**

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78(a)(2) or CFR 1.78(a)(4), and need not otherwise be made part of the specification.

Prior Application Status	Pending		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	
	a 371 of international	PCT/US2009/039695	2009-04-06	
Prior Application Status	Expired		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	
PCT/US2009/039695	non provisional of	61166651	2009-04-03	
Prior Application Status	Expired		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	
PCT/US2009/039695	non provisional of	61113471	2008-11-11	
Prior Application Status	Expired		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	
PCT/US2009/039695	non provisional of	61123438	2008-04-07	
Additional Domestic Benefit/National Stage Data may be generated within this form				

by selecting the **Add** button.

# **Foreign Priority Information:**

This section allows for the applicant to claim benefit of foreign priority and to identify any prior foreign application for which priority is not claimed. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(a).

		٩	<b>≀emove</b>		
Application Number	Country <sup>i</sup>	Parent Filing Date (YYYY-MM-DD)	Priority Claimed		
			🔿 Yes 💿 No		
Additional Foreign Priority Data may be generated within this form by selecting the Add button.					

# **Assignee Information:**

Providing this information in the application data sheet does not substitute for compliance with any requirement of part 3 of Title 37 of the CFR to have an assignment recorded in the Office.				
Assignee 1 Remove				
If the Assignee is an Organization check here.				
Organization Name Express Mobile Inc.				

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	XPR.002US0
		Application Number	
Title of Invention	SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES		

Mailing Address Information:					
Address 1	12 Pamaron Way	12 Pamaron Way			
Address 2	Suite 1	Suite 1			
City	Novato	Novato State/Province CA			
Country i US		Postal Code	94947		
Phone Number		Fax Number			
Email Address					
Additional Assignee Data may be generated within this form by selecting the Add Add					

# Signature:

A signature of the applicant or representative is required in accordance with 37 CFR 1.33 and 10.18. Please see 37 CFR 1.4(d) for the form of the signature.

Signature	e /Steven R. Vosen/		Date (YYYY-MM-DD)	2010-10-04	
First Name	Steven	Last Name	Vosen	Registration Number	45186

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450**.

# **Privacy Act Statement**

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

- The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
- 2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
- 3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
- 4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
- 5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
- 6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
- 7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
- 8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
- 9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

### SYSTEMS AND METHODS FOR PROGRAMMING MOBILE DEVICES

### TECHNICAL FIELD

[0001] The present invention generally relates to providing software for mobile devices, and more particularly to a method and system for authoring Applications for devices.

### BACKGROUND ART

[0002] Internet-connected mobile devices are becoming ever more popular. While these devices provide portability to the Internet, they generally do not have the capabilities of non-mobile devices including computing, input and output capabilities.

[0003] In addition, the mobility of the user while using such devices provides challenges and opportunities for the use of the Internet. Further, unlike non-mobile devices, there are a large number of types of devices and they tend to have a shorter lifetime in the marketplace. The programming of the myriad of mobile devices is a time-consuming and expensive proposition, thus limiting the ability of service providers to update the capabilities of mobile devices.

[0004] Thus there is a need in the art for a method and apparatus that permits for the efficient programming of mobile devices. Such a method and apparatus should be easy to use and provide output for a variety of devices.

### DISCLOSURE OF INVENTION

[0005] In certain embodiments, a system is provided to generate code to provide content on a display of a platform. The system includes a database of web services obtainable over a network and an authoring tool. The authoring tool is configured to define an object for presentation on the display, select a component of a web service included in said database, associate said object with said selected component, and produce code that, when executed on the platform, provides said selected component on the display of the platform.

[0006] In certain other embodiments, a method is provided for providing information to platforms on a network. The method includes accepting a first code over the network, where said first code is platform-dependent; providing a second code over the network, where said second code is platform-independent; and executing said first code and said second code on the platform to provide web components obtained over the network.

[0007] In certain embodiments, a method for displaying content on a platform utilizing a database of web services obtainable over a network is provided. The method includes: defining an object for presentation on the display; selecting a component of a web service included in said docket number: XPR.002WO0 1

database; associating said object with said selected component; and producing code that, when executed on the platform, provides said selected component on the display of the platform.

[0008] In one embodiment, one of the codes is a Player, which is a thin client architecture that operates in a language that manages resources efficiently, is extensible, supports a robust application model, and has no device specific dependencies. In another embodiment, Player **P** is light weight and extends the operating system and/or virtual machine of the device to: Manage all applications and application upgrades, and resolve device, operating system, VM and language fragmentation.

[0009] In another embodiment, one of the codes is an Application that is a device independent code that interpreted by the Player.

[0010] These features together with the various ancillary provisions and features which will become apparent to those skilled in the art from the following detailed description, are attained by the system and method of the present invention, preferred embodiments thereof being shown with reference to the accompanying drawings, by way of example only, wherein:

### BRIEF DESCRIPTION OF DRAWINGS

[0011] FIG. 1A is an illustrative schematic of one embodiment of a system including an authoring platform and a server for providing programming instructions to a device over a network;

[0012] FIG. 1B is schematic of an alternative embodiment system for providing programming instructions to device over a network;

[0013] FIG. 2A is a schematic of an embodiment of system illustrating the communications between different system components;

[0014] FIG.2B is a schematic of one embodiment of a device illustrating an embodiment of the programming generated by authoring platform;

[0015] FIGS. 3A and 3B illustrate one embodiment of a publisher interface as it appears, for example and without limitation, on a screen while executing an authoring tool;

[0016] FIG. 3C illustrates an embodiment of the Events Tab'

[0017] FIG. 3D illustrates one embodiment of an Animation Tab;

[0018] FIG. 3E illustrates one embodiment of Bindings Tab;

[0019] FIG. 3F illustrates one embodiment of a pop-up menu for adding web components;

[0020] FIG. 4A shows a publisher interface having a layout on a canvas; and FIG. 4B shows a device having the resulting layout on a device screen;

- [0021] FIG. 5 shows a display of launch strips;
- [0022] FIG. 6A is a display of a Channel Selection List;
- [0023] FIG. 6B is a display of a Widget Selection List;
- [0024] FIG. 6C is a display of a Phone List;
- [0025] FIG. 7 shows a display of a mash-up;
- [0026] FIG. 8 is a schematic of an embodiment of a push capable system;
- [0027] FIG. 9 is a schematic of an alternative embodiment of a push capable system;
- [0028] FIG. 10 is a schematic of one embodiment of a feed collector;
- [0029] FIG. 11 is a schematic of an embodiment of a Mobile Content Gateway;

[0030] FIG. 12 is a schematic of one embodiment of a system that includes a response director, a user agent database, an IP address database, and a file database; and

[0031] FIG. 13 is a schematic of another embodiment of a system that includes a response director, a user agent database, an IP address database, and a file database.

[0032] Reference symbols are used in the Figures to indicate certain components, aspects or features shown therein, with reference symbols common to more than one Figure indicating like components, aspects or features shown therein.

### MODE(S) FOR CARRYING OUT THE INVENTION

[0033] Figure 1A is an illustrative schematic of one embodiment of a system 100 including an authoring platform 110 and a server 120 for providing programming instructions to a device 130 over a network N. In one embodiment, device 130 is a wireless device, and network N includes wireless communication to the device. Alternatively, system 100 may provide access over network N to other information, data, or content, such as obtainable as a web service over the

Internet. In general, a user of authoring platform **110** may produce programming instructions or files that may be transmitted over network **N** to operate device 130, including instructions or files that are sent to device **130** and/or server **120**. The result of the authoring process is also referred to herein, and without limitation, as publishing an Application.

[0034] Embodiments include one or more databases that store information related to one or more devices **130** and/or the content provided to the devices. It is understood that such databases may reside on any computer or computer system on network **N**, and that, in particular, the location is not limited to any particular server, for example.

[0035] Device **130** may be, for example and without limitation, a cellular telephone or a portable digital assistant, includes a network interface **131**, a memory **133**, a processor **135**, a screen **137**, and an input device **139**. Network interface **131** is used by device **130** to communication over a wireless network, such as a cellular telephone network, a WiFi network or a WiMax network, and then to other telephones through a public switched telephone network (PSTN) or to a satellite, or over the Internet. Memory **133** includes programming required to operate device **130** (such as an operating system or virtual machine instructions), and may include portions that store information or programming instructions obtained over network interface **131**, or that are input by the user (such as telephone numbers or images from a device camera (not shown). In one embodiment screen **137** is a touch screen, providing the functions of the screen and input device **139**.

[0036] Authoring platform **110** includes a computer or computer system having a memory **111**, a processor **113**, a screen **115**, and an input device **117**. It is to be understood that memory **111**, processor **113**, screen **115**, and input device **117** are configured such a program stored in the memory may be executed by the processor to accept input from the input device and display information on the screen. Further, the program stored in memory **111** may also instruct authoring platform **110** to provide programming or information, as indicated by the line labeled "A" and to receive information, as indicated by the line labeled "B."

[0037] Memory 111 is shown schematically as including a stored program referred to herein, and without limitation, as an authoring tool 112. In one embodiment, authoring tool 112 is a graphical system for designing the layout of features as a display that is to appear on screen 137. One example of authoring tool 112 is the CDER<sup>TM</sup> publishing platform (Express Mobile, Inc., Novato, CA).

4

[0038] In another embodiment, which is not meant to limit the scope of the present invention, device **130** may include an operating system having a platform that can interpret certain routines. Memory **111** may optionally include programming referred to herein, and without limitation, as routines **114** that are executable on device **130**.

[0039] Routines **114** may include device-specific routines - that is, codes that are specific to the operating system, programming language, or platform of specific devices **130**, and may include, but are not limited to, Java, Windows Mobile, Brew, Symbian OS, or Open Handset Alliance (OHA). Several examples and embodiments herein are described with reference to the use of Java. It is to be understood that the invention is not so limited, except as provided in the claims, and that one skilled in the art could provide Players for devices using routines provided on a platform. Thus as an example, routines **114** may include Java API's and an authoring tool System Development Kit (SDK) for specific devices **130**.

[0040] Server 120 is a computer or computer system that includes a network interface 121, a memory 123.and a processor 125. Is to be understood that network interface 121, memory 123, and processor 125 are configured such that a program stored in the memory may be executed by the processor to: accept input and/or provide output to authoring platform 110; accept input and/or provide output through network interface 121 over network N to network interface 131; or store information from authoring platform 110 or from device 130 for transmission to another device or system at a later time.

[0041] In one embodiment, authoring platform **110** permits a user to design desired displays for screen **137** and actions of device **130**. In other words, authoring platform **110** is used to program the operation of device **130**. In another embodiment, authoring platform **110** allows a user to provide input for the design of one or more device displays and may further allow the user to save the designs as device specific Applications. The Applications may be stored in memory **123** and may then be sent, when requested by device **130** or when the device is otherwise accessible, over network **N**, through network interface **130** for storage in memory **133**.

[0042] In an alternative embodiment, analytics information from devices **130** may be returned from device **130**, through network **N** and server **120**, back to authoring platform **110**, as indicated by line **B**, for later analysis. Analytics information includes, but is not limited to, user demographics, time of day, and location. The type of analytic content is only limited by which listeners have been activated for which objects and for which pages. Analytic content may include, but is not limited to, player-side page view, player-side forms-based content, player-side user interactions, and player-side object status. Docket Number XPR.002WO0 5 [0043] Content server 140 is a computer or computer system that includes a network interface 141, a memory 143.and a processor 145. It is to be understood that network interface 141, memory 143, and processor 145 are configured such that a stored program in the memory may be executed by the processor to accepts requests **R** from device 130 and provide content **C** over a network, such as web server content the Internet, to device 130.

[0044] Figure 1B is schematic of an alternative embodiment system **100** for providing programming instructions to device **130** over a network **N** that is generally similar to the system of FIG. 1A. The embodiment of FIG. 1B illustrates that system **100** may include multiple servers **120** and/or multiple devices **130**.

[0045] In the embodiment of FIG. 1B, system **100** is shown as including two or more servers **120**, shown illustratively and without limitation as servers **120a** and **120b**. Thus some of the programming or information between authoring platform **110** and one or more devices **130** may be stored, routed, updated, or controlled by more than one server **120**. In particular, the systems and methods described herein may be executed on one or more server **120**.

[0046] Also shown in FIG. 1B are a plurality of devices 130, shown illustratively and without limitation as device 130-1, 130-1, ... 130-N. System 100 may thus direct communication between individual server(s) 120 and specific device(s) 130.

[0047] As described subsequently, individual devices **130** may be provided with program instructions which may be stored in each device's memory **133** and where the instructions are executed by each device's processor **135**. Thus, for example, server(s) **120** may provide device(s) **130** with programming in response to the input of the uses of the individual devices. Further, different devices **130** may be operable using different sets of instructions, that is having one of a variety of different "device platforms." Differing device platforms may result, for example and without limitation, to different operating systems, different versions of an operating system, or different versions of virtual machines on the same operating system. In some embodiments, devices **130** are provided with some programming from authoring system **100** that is particular to the device.

[0048] In one embodiment, system 100 provides permits a user of authoring platform 110 to provide instructions to each of the plurality of devices 130 in the form of a device- or device-platform specific instructions for processor 135 of the device, referred to herein and without limitation as a "Player," and a device-independent program, referred to herein and without limitation as an "Application" Thus, for example, authoring platform 110 may be used to

generate programming for a plurality of devices **130** having one of several different device platforms. The programming is parsed into instructions used by different device platforms and instructions that are independent of device platform. Thus in one embodiment, device **130** utilizes a Player and an Application to execute programming from authoring platform **110**. A device having the correct Player is then able to interpret and be programmed according to the Application.

[0049] In one alternative embodiment, the Player is executed the first time by device **130** ("activated") through an Application directory. In another alternative embodiment, the Player is activated by a web browser or other software on device **130**. In yet another alternative embodiment, Player is activated through a signal to device **130** by a special telephone numbers, such as a short code.

[0050] When the Application and the Player are provided to memory **133**, the functioning of device **130** may occur in accordance with the desired programming. Thus in one embodiment, the Application and Player includes programming instructions which may be stored in memory **133** and which, when executed by processor **135**, generate the designed displays on screen **137**. The Application and Player may also include programming instructions which may be stored in memory **133** and which provide instructions to processor **135** to accept input from input device **139**.

[0051] Authoring tool **112** may, for example, produce and store within memory **111** a plurality of Players (for different devices **130**) and a plurality of Applications for displaying pages on all devices. The Players and Applications are then stored on one or more servers **120** and then provided to individual devices **130**. In general, Applications are provided to device **130** for each page of display or a some number of pages. A Player need be provided once or updated as necessary, and thus may be used to display a large number of Applications. This is advantageous for the authoring process, since all of the device-dependent programming is provided to a device only once (or possibly for some small number of upgrades), permitting a smaller Application, which is the same for each device **130**.

[0052] Thus, for example and without limitation, in one embodiment, the Player transforms device-independent instructions of the Application into device-specific instructions that are executable by device **130**. Thus, by way of example and without limitation, the Application may include Java programming for generating a display on screen **137**, and the Player may interpret the Java and instruct processor **135** to produce the display according to the Application for execution on a specific device **130** according to the device platform. The Application may in Docket Number XPR.002WO0 7

general include, without limitation, instructions for generating a display on screen 137, instructions for accepting input from input device 139, instructions for interacting with a user of device 130, and/or instructions for otherwise operating the device, such as to place a telephone call.

[0053] The Application is preferably code in a device-independent format, referred to herein and without limitation as a Portable Description Language (PDL). The device's Player interprets or executes the Application to generate one or more "pages" ("Applications Pages") on a display as defined by the PDL. The Player may include code that is device-specific - that it, each device is provided with a Player that is used in the interpretation and execution of Applications. Authoring tool **112** may thus be used to design one or more device-independent Applications and may also include information on one or more different devices **130** that can be used to generate a Player that specific devices may use to generate displays from the Application.

[0054] In one embodiment, system **100** provides Players and Applications to one server **120**, as in FIG. 1A. In another embodiment, system **100** provides Players to a first server **120a** and Applications to a second server **120**b, as in FIG. 1B.

[0055] In one embodiment, authoring tool **112** may be used to program a plurality of different devices **130**, and routines **114** may include device-specific routines. In another embodiment, the Player is of the type that is commonly referred to as a "thin client" – that is, software for running on the device as a client in client-server architecture with a device network which depends primarily on a central server for processing activities, and mainly focuses on conveying input and output between the user and the server.

[0056] In one embodiment, authoring platform **110** allows user to arrange objects for display on screen. A graphical user interface ("GUI," or "UI") is particularly well suited to arranging objects, but is not necessary. The objects may correspond to one or more of an input object, an output object, an action object, or may be a decorative display, such as a logo, or background color or pattern, such as a solid or gradient fill. In another embodiment, authoring platform **110** also permits a user to assign actions to one or more of an input object, an output object, or an action object. In yet another embodiment, authoring platform **110** also permits a user to bind one or more of an input object, an output object, or an action object with web services or web components, or permits a user to provide instructions to processor **135** to store or modify information in memory **133**, to navigate to another display or service, or to perform other actions, such as dialing a telephone number.

8

[0057] In certain embodiments, the applicant model used in developing and providing Applications is a PDL. The PDL can be conceptually viewed as a device, operating system and virtual machine agnostic representation of Java serialized objects. In certain embodiments, the PDL is the common language for authoring tool **112**, the Application, and Player. Thus while either designing the Application with the authoring tool **112**, or programming with the SDK, the internal representation of the programming logic is in Java. In one embodiment the SDK is used within a multi-language software development platform comprising an IDE and a plug-in system to extend it, such as the Eclipse Integrated Development Environment (see, for example, http://www.eclipse.org/). At publish time the Java code is translated into a PDL. This translation may also occur in real-time during the execution of any Web Services or backend business logic that interacts with the user.

[0058] One embodiment for compacting data that may be used is described in co-pending U.S. Patent No. 6,546,397 to Rempell ("Rempell"), the contents of which are incorporated herein by reference. In that patent the compressed data is described as being a database. The terminology used here is a PDL, that is the "internal database" of Rempell is equivalent to the PDL of the present Application.

[0059] The use of a PDL, as described in Rempell, permits for efficient code and data compaction. Code, as well as vector, integer and Boolean data may be compacted and then compressed resulting in a size reduction of 40 to 80 times that of the original Java serialized objects. This is important not only for performance over the network but for utilizing the virtual memory manager of the Player more efficiently. As an example, the reassembled primitives of the Java objects may first undergo logical compression, followed by LZ encoding.

[0060] The use of a PDL also provides virtual machine and operating system independence. Since the reassembled primitives of the Application no longer have any dependencies from the original programming language (Java) that they were defined in. The PDL architecture takes full advantage of this by abstracting all the virtual machine and/or operating system interfaces from the code that processes the PDL.

[0061] In one embodiment, the PDL is defined by the means of nested arrays of primitives. Accordingly, the use of a PDL provides extensibility and compatibility, with a minimal amount of constraints in extending the Player seamlessly as market demands and device capabilities continue to grow. Compatibility with other languages is inherent based on the various Player abstraction implementations, which may be, for example and without limitation, Java CDC, J2SE or MIDP2 implementations.

Docket Number XPR.002WO0

9

[0062] In one embodiment, the architecture of Player **P** includes an abstraction interface that separates all device, operating system and virtual machine dependencies from the Player's Application model business logic (that is, the logic of the server-side facilities) that extend the Application on the Player so that it is efficiently integrated into a comprehensive client/server Application. The use of an abstraction interface permits the more efficient porting to other operating systems and virtual machines and adding of extensions to the Application model so that a PDL can be implemented once and then seamlessly propagated across all platform implementations. The Application model includes all the currently supported UI objects and their attributes and well as all of the various events that are supported in the default Player. Further, less robust platforms can be augmented by extending higher end capabilities inside that platform's abstraction interface implementation.

[0063] In one embodiment, authoring platform **110** provides one or more pages, which may be provided in one Application, or a plurality of Applications, which are stored in memory **123** and subsequently provided to memory **133**. In certain embodiments, the Application includes instructions **R** to request content or web services **C** from content server **140**. Thus, for example and without limitation, the request is for information over the network via a web service, and the request **R** is responded to with the appropriate information for display on device **130**. Thus, for example, a user may request a news report. The Application may include the layout of the display, including a space for the news, which is downloaded form content server **140** for inclusion on the display. Other information that may be provided by content server **140** may include, but is not limited to, pages, Applications, multimedia, and audio.

[0064] Figure 2A is a schematic of a system 200 of an embodiment of system 100 illustrating the communications between different system components. System includes a response director 210, a web component registry 220, and a web service 230. System 200 further includes authoring platform 110, server 120, device 130 and content server 140 are which are generally similar to those of the embodiments of FIGS. 1A and 1B, except as explicitly noted.

[0065] Response director **210** is a computer or computer system that may be generally similar to server **120** including the ability to communicate with authoring platform **110** and one or more devices **130**. In particular, authoring platform **110** generates one or more Players (each usable by certain devices **130**) which are provided to response director **210**. Devices **130** may be operated to provide response director **210** with a request for a Player and to receive and install the Player. In one embodiment, device **130** provides response director **210** with device-specific information including but not limited to make, model, and/or software version of the device. Response

director **210** then determines the appropriate Player for the device, and provides the device with the Player over the network.

[0066] Web service 230 is a plurality of services obtainable over the Internet. Each web service is identified and/or defined as an entry in web component registry 230, which is a database, XML file, or PDL that exists on a computer that may be a server previously described or another server 120. Web component registry 230 is provided through server 120 to authoring platform 110 so that a user of the authoring platform may bind web services 230 to elements to be displayed on device 130, as described subsequently.

[0067] In one embodiment, authoring platform **110** is used in conjunction with a display that provides a WYSIWYG environment in which a user of the authoring platform can produce an Application and Player that produces the same display and the desired programming on device **130**. Thus, for example, authoring tool **112** provides a display on screen **115** that corresponds to the finished page that will be displayed on screen **137** when an Application is intercepted, via a Player, on processor **135** of device **130**.

[0068] Authoring platform **110** further permits a user of the authoring platform to associate objects, such as objects for presenting on screen **137**, with components of one or more web services **230** that are registered in web component registry **220**. In one embodiment, information is provided in an XML file to web component registry **220** for each registered components of each web service **230**. Web component registry **220** may contain consumer inputs related to each web service **230**, environmental data such as PIM, time or location values, persistent variable data, outputs related to the web service, and/or optional hinting for improving the user's productivity.

[0069] A user of authoring platform **110** of system **200** may define associations with web services as WebComponent Bindings. In one embodiment, authoring platform **110** allows a user to associate certain objects for display that provide input or output to components of web service **230**. The associated bindings are saved as a PDL in server **120**.

[0070] In one embodiment, an XML web component registry 220 for each registered web service 230 is loaded into authoring platform 110. The user of system 200 can then assign components of any web service 230 to an Application without any need to write code. In one embodiment, a component of web service 230 is selected from authoring platform 110 which presents the user with WYSIWYG dialog boxes that enable the binding of all the inputs and outputs of component of web service 230 to a GUI component of the Application as will be

displayed on screen 137. In addition, multiple components of one or more web service 230 can be assigned to any Object or Event in order to facilitate mashups. These Object and/or Event bindings, for each instance of a component of any web service 230, are stored in the PDL. The content server 140 handles all communication between device 130 and the web service 230 and can be automatically deployed as a web application archive to any content server.

[0071] Device 130, upon detecting an event in which a component of a web service 230 has been defined, assembles and sends all related inputs to content server 240, which proxies the request to web service 230 and returns the requested information to device 130. The Player on device 130 then takes the outputs of web service 230 and binds the data to the UI components in the Application, as displayed on screen 137.

[0072] In one embodiment, the mechanism for binding the outputs of the web service to the UI components is through symbolic references that matches each output to the symbolic name of the UI component. The outputs, in one embodiment, may include meta-data which could become part of the inputs for subsequent interactions with the web service.

[0073] For example, if a user of authoring platform **110** wants to present an ATOM feed on device **130**, they would search through a list of UI Components available in the authoring platform, select the feed they want to use, and bind the output of the feed summary to a textbox. The bindings would be saved into the PDL on server **120** and processed by device **130** at runtime. If the ATOM feed does not exist a new one can be added to the web component registry that contains all the configuration data required, such as the actual feed URL, the web component manager URL, and what output fields are available for binding.

[0074] In another embodiment, components of web services **230** are available either to the user of authoring platform **110** or otherwise accessible through the SDK and Java APIs of routines **114**. System **200** permits an expanding set of components of web services **230** including, but not limited to: server pages from content server **120**; third-party web services including, but not limited to: searching (such through Google or Yahoo), maps (such as through MapQuest and Yahoo), storefronts (such as through ThumbPlay), SMS share (such as through clickatel), stock quotes, social networking (such as through FaceBook), stock quotes, weather (such as through Accuweather) and/or movie trailers. Other components include web services for communication and sharing through chats and forums and rich messaging alerts, where message alerts are set-up that in turn could have components of Web Services **230** defined within them, including the capture of consumer generated and Web Service supplied rich media and textual content.

[0075] System **200** also permits dynamic binding of real-time content, where the inputs and outputs of XML web services are bound to GUI components provided on screen **137**. Thus, for example, a user of authoring platform **110** may bind attributes of UI Objects to a particular data base field on a Server. When running the Application, the current value in the referenced data base will be immediately applied. During the Application session, any other real time changes to these values in the referenced data base will again be immediately displayed.

[0076] As an example of dynamic binding of real-time content, an RSS feeds and other forms of dynamic content may be inserted into mobile Applications, such as device **130**, using system **200**. Authoring platform **110** may include a "RSS display" list which permits a user to select RSS channels and feeds from an extensible list of available dynamic content. Meta data, such as titles, abstracts and Images can be revealed immediately by the user as they traverse this RSS display list, bringing the PC experience completely and conveniently to mobile devices **130**. In addition, Authoring platform **110** may include a dialog box that dynamically links objects to data and feeds determined by RSS and chat databases. Any relevant attribute for a page view and/or object can be dynamically bound to a value in a server-side database. This includes elements within complex objects such as: any icon or text element within a graphical list; any icon within a launch strip; any feature within any geographical view of a GIS service object; and/or any virtual room within a virtual tour.

[0077] As an example of third-party web services 230 that may be provided using system 200, a user of authoring platform 110 can place, for example, Yahoo maps into device 130 by binding the required component of the Yahoo Maps Web Service, such as Yahoo Map's Inputs and/or Outputs to appropriate Objects of authoring platform 110. System 200 also provides binding to web services for text, image and video searching by binding to components of those web services.

[0078] In one embodiment, an Application for displaying on device **130** includes one or more Applications Pages, each referred to herein as an "XSP," that provides functionality that extends beyond traditional web browsers. The XSP is defined as a PDL, in a similar manner as any Application, although it defines a single page view, and is downloaded to the Player dynamically as required by the PDL definition of the Application. Thus, for example, while JSPs and ASPs, are restricted to the functionality supported by the web browser, the functionality of XSPs can be extended through authoring platform **110** having access to platform dependent routines **114**, such as Java APIs. Combined with dynamic binding functionality, an XSP, a page can be saved as a page object in an author's "pages" library, and then can be dynamically populated with real-time

Docket Number XPR.002WO0

13

content simultaneously as the page is downloaded to a given handset Player based on a newly expanded API. XSP Server Pages can also be produced programmatically, but in most cases authoring platform **110** will be a much more efficient way to generate and maintain libraries of dynamically changing XSPs.

[0079] With XSPs, Applications Pages that have dynamic content associated with them can be sent directly to device **130**, much like how a web browser downloads an HTML page through a external reference. Without XSPs, content authors would have to define each page in the Application. With XSPs, no pages need to be defined. Thus, for example, in a World Cup Application, one page could represent real-time scores that change continuously on demand. With polling (for example, a prompt to the users asking who they predict will win a game), a back-end database would tabulate the information and then send the results dynamically to the handsets. With a bar chart, the Application would use dynamic PDL with scaling on the fly. For example, the server would recalibrate the bar chart for every ten numbers.

[0080] Other combinations of components of web services 230 include, but are not limited to, simultaneous video chat sessions, inside an integrated page view, with a video or television station; multiple simultaneous chat sessions, each with a designated individual and/or group, with each of the chat threads visible inside an integrated page view.

[0081] Another extension of an XSP is a widget object. Widgets can be developed from numerous sources including, but not limited to, authoring platform 110, a Consumer Publishing Tool, and an XML to Widget Conversion Tool where the SDK Widget Libraries are automatically populated and managed, or Widget Selection Lists that are available and can be populated with author defined Icons.

### APPLICATIONS, PLAYERS, AND PROCESSING IN A DEVICE

[0082] Figure 2B is a schematic of one embodiment of a device **130** illustrating an embodiment of the programming generated by authoring platform **110**. Memory **133** may include several different logical portions, such as a heap **133a**, a record store **133b** and a filesystem (not shown).

[0083] As shown in FIG. 2B, heap **133a** and record store **133b** include programming and/or content. In general, heap **133a** is readily accessible by processor **135** and includes, but is not limited to portions that include the following programming: a portion **133a1** for virtual machine compliant objects representing a single Page View for screen **137**; a portion **133a2** for a Player; a portion **133a3** for a virtual machine; and a portion **133a4** for an operating system. Docket Number XPR.002WO0 14

[0084] Record store 133b (or alternatively the filesystem) includes, but is not limited to, portions 133b1 for Applications and non-streaming content, which may include portions 133a2 for images, portions 133a4 for audio, and/or portions 133a5 for video. and portions 133b2 for non-Application PDLs, such as a Master Page PDL for presenting repeating objects, and Alerts, which are overlayed on the current page view. Other content, such as streaming content may be provided from network interface 131 directly to the Media Codec of device 130 with instructions from Player on how to present the audio or video.

[0085] In one embodiment, the Player includes a Threading Model and a Virtual Memory Manager. The Threading Model first manages a queue of actions that can be populated based on Input/Output events, Server-side events, time-based events, or events initiated by user interactions. The Threading Model further manages the simultaneous execution of actions occurring at the same time. The Virtual Memory Manager includes a Logical Virtual Page controller that provides instructions from the record store to the heap, one page at time. Specifically, the Virtual Memory Manager controls the transfer of one of the Application Pages and its virtual machine compliant objects into portion **133a1** as instructions readable by the Player or Virtual Machine. When the Player determines that a new set of instructions is required, the information (such as one Application Page is retrieve from the Record store, converted into virtual machine compliant objects (by processor **135** and according to operation by the Player, Virtual Machine, etc). and stored in heap **133a**. Alternatively, the Player may augment virtual machine compliant objects with its own libraries for managing user interactions, events, memory, etc.

[0086] The connection of portions 133a1, 133a2, 133a3, 133a4, record store 133b and processor 135 are illustrative of the logical connection between the different types of programming stored in Heap 133a and record store 133b, that is, how data is processed by processor 135.

[0087] The Player determines which of the plurality of Application Pages in portion **133b1** is required next. This may be determined by input actions from the Input Device **139**, or from instructions from the current Application Page. The Player instructs processor **135** to extract the PDF from that Applications Page and store it in portion **133a1**. The Player then interprets the Application Page extracted from PDL which in turn defines all of the virtual machine compliant Objects, some of which could have attributes that refer to images, audio, and/or video stored in portions **133a3**, **133a4**, **133a5**, respectively.

[0088] The Virtual Machine in portion **133a3** processes the Player output, the Operating System in portion **133a3** processes the Virtual Machine output which results in machine code that is processed by the Operating System in portion **133a4**.

[0089] In another embodiment, the Player is a native program that interacts directly with the operating system.

### EMBODIMENTS OF A PUBLISHING ENVIRONMENT

[0090] In one embodiment, authoring platform **110** includes a full-featured authoring tool **112** that provides a what-you-see-is-what-you-get (WYSIWYG) full featured editor. Thus, for example, authoring tool **112** permits a user to design an Application by placing objects on canvas **305** and optionally assigning actions to the objects and save the Application. System **100** then provides the Application and Player to a device **130**. The Application as it runs on device **130** has the same look and operation as designed on authoring platform **110**. In certain embodiments, authoring platform **110** is, for example and without limitation, a PC-compatible or a Macintosh computer.

[0091] Authoring platform **110** produces an Application having one or more Applications Pages, which are similar to web pages. That is, each Applications Page, when executed on device **130** may, according to its contents, modify what is displayed on screen **137** or cause programming on the device to change in a manner similar to how web pages are displayed and navigated through on a website.

[0092] In one embodiment, authoring tool **112** allows a user to place one or more objects on canvas **305** and associate the objects with an Applications Pages. Authoring platform **110** maintains a database of object data in memory **111**, including but not limited to type of object, location on which page, and object attributes. The user may add settings, events, animations or binding to the object, from authoring tool **112**, which are also maintained in memory **111**. Authoring tool **112** also allows a user to define more than one Applications Page.

[0093] In another embodiment, authoring tool **112**, provides Java programming functions of the Java API for specific devices **130** as pull-down menus, dialog boxes, or buttons. This permits a user of authoring platform **110** to position objects that, after being provided as an Application to device **130**, activate such Java functions on the device.

[0094] In certain embodiments, authoring platform **110**, as part of system **100**, permits designers to include features of advanced web and web services Applications for access by users

of device **130**. Some of the features of advanced web and web services include, but are not limited to: slide shows, images, video, audio, animated transitions, multiple chats, and mouse interaction; full 2-D vector graphics; GIS (advanced LBS), including multiple raster and vector layers, feature sensitive interactions, location awareness, streaming and embedded audio/video, virtual tours, image processing and enhancement, and widgets. In other embodiments the features are provided for selection in authoring platform **110** through interactive object libraries.

[0095] In certain embodiments, authoring platform **110**, as part of system **100**, allows the inclusion of child objects which may eventually be activated on device **130** by the user of the device or by time. The uses of the child objects on device **130** include, but are not limited to: mouse over (object selection), hover and fire events and launching of object-specific, rich-media experiences.

[0096] In certain other embodiments, authoring platform **110**, as part of system **100**, provides advanced interactive event models on device **130**, including but not limited to: user-, time- and/or location-initiated events, which allow content developers to base interactivity on specific user interactions and/or instances in time and space; timelines, which are critical for timing of multiple events and for animations when entering, on, or exiting pages of the Application; waypoints, which act similar to key frames, to allow smooth movement of objects within pages of the Application. Waypoints define positions on a page object's animation trajectory. When an object reaches a specific waypoint other object timelines can be initiated, creating location-sensitive multiple object interaction, and/or audio can be defined to play until the object reaches the next waypoint.

[0097] Authoring platform **110** may also define a Master Page, which acts as a template for an Applications Page, and may also define Alert Pages, which provide user alerts to a user of device **130**.

[0098] In certain embodiments, authoring platform **110**, as part of system **100**, provides full style inheritance on device **130**. Thus, for example and without limitation, both master page inheritance (for structural layout inheritance and repeating objects) and object styles (for both look and feel attribute inheritance) are supported. After a style has been defined for an object, the object will inherit the style. Style attributes include both the look and the feel of an object, including mouse interaction, animations, and timelines. Each page may include objects that may be a parent object or a child object. A child object is one that was created by first selecting a parent object, and then creating a child object. Child objects are always part of the same drawing layer as its parent object, but are drawn first, and are not directly selectable when running the Docket Number XPR.002WO0 17

Application. A parent object is any object that is not a child object, and can be selected when running the Application.

[0099] As an example, the user of authoring tool **112** may create various text objects on canvas **305** using a style that sets the font to red, the fonts of these objects will be red. Suppose user of authoring tool **112**changes the font color of a specific button to green. If later, the user of authoring tool **112**changes the style to blue; all other text objects that were created with that style will become blue except for the button that had been specifically set to green.

[0100] In certain other embodiments, authoring platform **110** provides page view, style, object, widget and Application template libraries. Authoring platform **110** may provide templates in private libraries (available to certain users of the authoring platform) and public libraries (available to all users of the authoring platform). Templates may be used to within authoring platform **110** to define the look and feel of the entire Application, specific pages, or specific slide shows and virtual tours a seen on device 130.

[0101] Figures 3A and 3B illustrate one embodiment of a publisher interface 300 as it appears, for example and without limitation, on screen 115 while executing authoring tool 112. In one embodiment, publisher interface 300 includes a Menu bar 301, a Tool bar 303, a Canvas 305, a Layer Inspector 307 having subcomponents of a page/object panel 307a, an object style panel 307b, and a page alert panel 307c, and a Resource Inspector 309.

[0102] In general, publisher interface **300** permits a user of authoring platform **110** to place objects on canvas **305** and then associate properties and/or actions to the object, which are stored in the Application. As described subsequently, publisher interface **300** permits a user to program a graphical interface for the screen **137** of device **130** on screen **115** of authoring platform **110**, save an Application having the programming instructions, and save a Player for the device. The intended programming is carried out on device **130** when the device, having the appropriate device platform Player, receives and executes the device-independent Application.

[0103] Thus, for example, authoring tool **112** maintains, in memory **111**, a list of every type of object and any properties, actions, events, or bindings that may be assigned to that object. As objects are selected for an Application, authoring tool **112** further maintains, in memory **111**, a listing of the objects. As the user selects objects, publisher interface **300** provides the user with a choice of further defining properties, actions, events, or bindings that may be assigned to each particular object, and continues to store the information in memory **111**.

[0104] In one embodiment, publisher interface **300** is a graphical interface that permits the placement and association of objects in a manner typical of, for example, vector graphics editing programs (such as Adobe Illustrator). Objects located on canvas **305** placed and manipulated by the various commands within publisher interface **300** or inputs such as an input device **117** which may be a keyboard or mouse. As described herein, the contents of canvas **305** may be saved as an Application that, through system **100**, provide the same or a similar placement of objects on screen **137** and have actions defined within publisher interface **300**. Objects placed on canvas **305** are intended for interaction with user of device **130** and are referred to herein, without limitation, as objects or UI (user interface) objects. In addition, the user of interface **300** may assign or associate actions or web bindings to UI objects placed on canvas **305** with result in the programming device **130** that cause it to respond accordingly.

[0105] Objects include, but are not limited to input UI objects, response UI objects. Input UI objects include but are not limited to: text fields (including but not limited to alpha, numeric, phone number, or SMS number); text areas; choice objects (including but not limited to returning the selected visible string or returning a numeric hidden attribute); single item selection lists (including but not limited to returning the selected visible string or returning a numeric hidden attribute); multi item selection lists (including but not limited to returning all selected items (visible text string or hidden attribute) or cluster item selection lists (returning the hidden attributes for all items).

[0106] Other input UI objects include but are not limited to: check boxes; slide show (including but not limited to returning a numeric hidden attribute, returning a string hidden attribute, or returning the hidden attributes for all slides); and submit function (which can be assigned to any object including submit buttons, vectors, etc.).

[0107] Response UI Objects may include, but are not limited to: single line text objects, which include: a text Field (including but not limited to a URL, audio URL, or purchase URL), a text button, a submit button, or a clear button. Another response UI objects include: a multiple line text object, which may include a text area or a paragraph; a check box; an image; a video; a slide show (with either video or image slides, or both); choice objects; list objects; or control lists, which control all the subordinate output UI objects for that web component. Control list objects include, but are not limited to: list type or a choice type, each of which may include a search response list or RSS display list.
[0108] As a further example of objects that may be used with authoring tool **112**, Table I lists Data Types, Preferred Input, Input Candidates, Preferred Output and Output Candidates for one embodiment of an authoring tool.

Data Types	Preferred Input	Input Candidates	Preferred Output	Output Candidates
boolean	Check Box	Check Box	Check Box	Check Box
Int	Text Field (integer)	Text Field (integer) Text Field (Phone #) Text Field (SMS #) Choice List (single select)	Text Field (integer)	Text Field (integer) Text Field (Phone #) Text Field (SMS #) Choice List (single select) Text Button
String	Text Field (Alpha)	Anv	Text Field (Alpha)	Any
multilineString	Text Area	Text Area	Text Area	Text Area Paragraph
ImageURL	N/A	N/A	Image	Image Slide Show
VideoURL	N/A	N/A	Video	Video Slide Show
List	Single Item List	Single Item List Multi-Select List Complex List Choice Slide Show	Single Item List	Any List Type Any Choice Type (see Complex List Specification)
ComplexList	Complex List	Single Item List Multi-Select List Complex List	Single Item List	Any List Type (see Complex List Specification)
Slideshow	Slide Show	Slide Show	Slide Show	Slide Show
SearchResponseList	N/A	N/A	Search Response List	Search Response List Control List Complex List Choice
RSSList	N/A	N/A	RSS Display List	RSS Display List Control List Complex List Choice
SingleSelectionList	Choice	Choice Complex List	Choice	Choice Complex List
MultiSelectionList	Multi-Selection List	Multi-Selection List	Multi-Selection List	Multi-Selection List
ServiceActivation	Submit Button	Any	N/A	N/A
ChannellmageURL	N/A	N/A	Image	lmage Video Slide Show
ChannelDescription	N/A	N/A	Text Area	Text Area
				Paragraph Text Field Text Button List Choice

ChannelTitle	N/A	N/A	Text Field	Text Field
				Text Button
				Paragraph
				Text Area
				List
				Choice
URL			Text Field	Text Field
			(URL request)	(URL request)
Audio URL			Text Field	Text Field
			(Audio URL request)	(Audio URL request)
Purchase URL			Text Field	Text Field
			(Purchase URL request)	(Purchase URL request)
Image Data			Image	Image
				Slide Show
Image List Data			Slide Show	Slide Show
				Image
Persistent Variable	N/A	N/A	N/A	N/A
Pipeline Multiple Select	Multi-select List	Multi-select List	N/A	N/A
		Complex List		
		Slide Show		
Phone Number	Text Field	Text Field	Text Field	Text Field
	(numeric type)	Text Button	(numeric type)	Text Button
Hidden Attribute	Complex List	Complex List	Complex List	Complex List
		Slide Show		Slide Show
Collection List	N/A	N/A	Slide Show	Complex List
				Slide Show
	1	1	1	

Table I. One embodiment of supported objects

[0109] In general, publisher interface **300** permits a user to define an Application as one or more Applications Pages, select UI objects from Menu bar **301** or Tool bar **303** and arrange them on an Applications Page by placing the objects canvas **305**. An Application Page is a page that is available to be visited through any navigation event. Application Pages inherit all the attributes of the Master Page, unless that attribute is specifically changed during an editing session.

[0110] Authoring platform **110** also stores information for each UI object on each Application Page of an Application. Layer Inspector **307** provides lists of Applications Pages, UI objects on each Applications Page, and Styles, including templates. Objects may be selected from canvas **305** or Layer Inspector **307** causing Resource Inspector **309** to provide lists of various UI objects attributes which may be selected from within the Resource Inspector. Publisher interface **300** also permits a user to save their work as an Application for layer transfer and operation of device **130**. Publisher interface **300** thus provides an integrated platform for designing the look and operation of device **130**.

[0111] The information stored for each UI object depends, in part, on actions which occur as the result of a user of device **130** selecting the UI object from the device. UI objects include, but Docket Number XPR.002WO0 21

are not limited to: navigational objects, such as widget or channel launch strips or selection lists; message objects for communicating, such as a multiple chat, video chat, phone and/or SMS lists or fields or a pop-up alert; text fields or areas; check boxes; pull down menus; selection lists and buttons; pictures; slide shows; video or LBS maps; shapes or text defined by a variety of tools; a search response; or an RSS display.

[0112] In certain embodiments, publisher interface **300** permits a user to assign action to UI objects, including but not limited to, programming of the device **130** or a request for information over network **N**. In one embodiment, for example and without limitation, publisher interface **300** has a selection to bind a UI object to a web service - that is, associate the UI object or a manipulation or selection of UI object with web services. Publisher interface **300** may also include many drawing and text input functions for generating displays that may be, in some ways, similar to drawing and/or word processing programs, as well as toolbars and for zooming and scrolling of a workspace.

[0113] Each UI object has some form, color, and display location associate with it. Further, for example and without limitation, UI objects may have navigational actions (such as return to home page), communications actions (such as to call the number in a phone number field), or web services (such as to provide and/or retrieve certain information from a web service). Each of the these actions requires authoring platform **110** to store the appropriate information for each action. In addition, UI objects may have associated patent or child objects, default settings, attributes (such as being a password or a phone number), whether a field is editable, animation of the object, all of which may be stored by authoring platform **110**, as appropriate.

[0114] Menu bar 301 provides access features of publisher interface 300 through a series of pull-down menus that may include, but are not limited to, the following pull-down menus: a File menu 301a, an Edit menu 301b, a View menu 301c, a Project menu 301d, an Objects menu 301e, an Events menu 301f, a Pages menu 301g, a Styles menu 301h, and a Help menu 301i.

[0115] File menu **301a** provides access to files on authoring platform **110** and may include, for example and without limitation, selections to open a new Application or master page, open a saved Application, Application template, or style template, import a page, alert, or widget, open library objects including but not limited to an image, video, slide show, vector or list, and copying an Application to a user or to Server **120**.

[0116] Edit menu **301b** may include, but is not limited to, selections for select, cut, copy, paste, and edit functions.

[0117] View menu **301c** may include, but is not limited to, selections for zooming in and out, previewing, canvas **305** grid display, and various palette display selections.

[0118] Project menu **301d** may include, but is not limited to, selections related to the Application and Player, such as selections that require a log in, generate a universal Player, generate server pages, activate server APIs and extend Player APIs. A Universal Player will include all the code libraries for the Player, including those that are not referenced by the current Application. Server APIs and Player APIs logically extend the Player with Server-side or deviceside Application specific logic.

[0119] Objects menu **301e** includes selections for placing various objects on canvas **305** including, but not limited to: navigation UI objects, including but not limited to widget or channel launch strips or selection lists; message-related UI objects, including but not limited to multiple chat, video chat, phone and/or SMS lists or fields, or a pop-up alert; shapes, which provides for drawing tools; forms-related objects, including but not limited to text fields; scrolling text box, check box, drop-down menu, list menu, submit button or clear button; media-related UI objects such as pictures, slide shows, video or LBS maps; text-related UI objects such as buttons or paragraphs; and variables, including but not limited to time, date and audio mute control.

[0120] Events menu **301f** includes selections for defining child objects, mouse events, animations or timelines.

[0121] Pages menu **301g** includes selection for handling multi-page Applications, and may include selections to set a master page, delete, copy, add or go to Applications Pages.

[0122] Styles menu **301h** includes selections to handle styles, which are the underlying set of default appearance attributes or behaviors that define any object that is attached to a style. Styles are a convenient way for quickly creating complex objects, and for changing a whole collection of objects by just modifying their common style. Selections of Styles menu **301h** include, but not limited to, define, import, or modify a style, or apply a template. Help menu **301i** includes access a variety of help topics.

[0123] Tool bar **303** provides more direct access to some of the features of publisher interface **300** through a series of pull-down menus. Selections under tool bar **303** may include selections to:

control the look of publisher interface 300, such as a Panel selection to control the
 Docket Number XPR.002WO0 23

for hiding or viewing various panels on publisher interface 300;

[0124] control the layout being designed, such as an Insert Page selection to permit a user to insert and name pages;

[0125] control the functionality of publisher interface **300**, such as a Palettes selection to choose from a variety of specialized palettes, such as a View Palette for zooming and controlling the display of canvas **305**, a Command Palette of common commands, and Color and Shape Palettes;

[0126] place objects on canvas **305**, which may include selections such as: a Navigation selection to place navigational objects, such as widget or channel launch strips or selection lists), a Messages selection to place objects for communicating, such as a multiple chat, video chat, phone and/or SMS lists or fields, or a pop-up alert, a Forms selection to place objects such as text fields or areas, check boxes, pull down menus, selection lists, and buttons, a Media selection to place pictures, slide shows, video or LBS maps, and a Shapes selection having a variety of drawing tools, a Text selection for placing text, a search response, or an RSS display, and Palettes.

[0127] In one embodiment, Tool bar 303 includes a series of pull-down menus that may include, but are not limited to, items from Menu bar 301 organized in the following pull-down menus: a Panel menu 303a, an Insert Page menu 303b, a Navigation menu 303c, a Messages menu 303d, a Forms menu 303e, a Media menu 303f, a Shapes menu 303g, a Text menu 303h, and a Palettes menu 301i.

[0128] Panel menu **303a** permits a user of authoring platform **110** to change the appearance of interface **300** by, controlling which tools are on the interface or the size of canvas **305**. Insert Page menu **303b** permits a user of authoring platform **110** to open a new Application Page. Navigation menu **303c** displays a drop down menu of navigational-related objects such as a widget or channel launch strip or selection list. Messages menu **303d** displays a drop down menu of messaging-related objects such as multiple chat, video chat, phone or SMS lists or fields, and pop-up alerts. Forms menu **303e** displays a drop down menu of forms-related objects including, but not limited to, a text field, a text area, a check box, a drop down menu, a selection list, a submit button, and a clear button. Media menu **303f** displays a drop down menu of media-related objects including, but not limited to, a picture, slide show, video or LBS map. Shapes menu **303g** displays a drop down menu of draw tools, basic shapes, different types of lines and arrows and access to a shape library. Text menu **303j** displays a drop down menu of text-related objects,

including but not limited to a text button, paragraph, search response, RSS display and variables such as time and date.

[0129] Palettes menu **301i** includes a selection of different palettes that can be moved about publisher interface **300**, where each palette has specialized commands for making adjustments or associations to objects easier. Palettes include, but are not limited to: a page view palette, to permit easy movement between Applications Pages; a view palette, to execute an Application or zoom or otherwise control the viewing of an Application; a commands palette having editing commands; a color palette for selection of object colors; and a shapes palette to facilitate drawing objects.

[0130] Layer inspector **307** permits a user of publisher interface **300** to navigate, select and manipulate UI objects on Applications Pages. Thus, for example, a Page/objects panel **307a** of layer inspector **307** has a listing that may be selected to choose an Applications Pages within and Application, and UI objects and styles within an Applications Page. An Object styles panel **307b** of layer inspector **307** displays all styles on the Applications Page and permits selection of UI objects for operations to be performed on the objects.

[0131] Thus, for example, when objects from Menu bar 301 or Tool bar 303 are placed on canvas 305, the name of the object appears in Page/objects panel 307a. Page/objects panel 307a includes a page display 307a1 and an objects display 307a2. Page display 307a1 includes a pull down menu listing all Applications Pages of the Application, and objects display 307a2 includes a list of all objects in the Applications Page (that is, objects on canvas 305).

[0132] In general, page/objects panel **307a** displays various associations with a UI object and permits various manipulations including, but not limited to, operations for parent and child objects that are assigned to a page, and operations for object styles, and permits navigating between page types and object styles, such as switching between the master page and Application pages and deselecting object styles and alerts, opening an Edit Styles Dialog Box and deselecting any master, Application or alert page, or selecting an alert page and deselecting any Master Page or Application Page. A parent or child object can also be selected directly from the Canvas. In either case, the Resource Inspector can then be used for modifying any attribute of the selected object.

[0133] Examples of operations provided by page/objects panel **307a** on pages include, but are not limited to: importing from either a user's private page library or a public page library; deleting a page; inserting a new page, inheriting all the attributes of the Master Page, and placing

the new page at any location in the Page List; editing the currently selected page, by working with an Edit Page Dialog Box. While editing all the functions of the Resource Inspector **309** are available, as described subsequently, but are not applied to the actual page until completing the editing process.

[0134] Examples of operations provided by of page/objects panel **307a** on objects, which may be user interface (UI) objects, include but are not limited to: changing the drawing order layer to: bring to the front, send to the back, bring to the front one layer, or send to the back one layer; hiding (and then reshowing) selected objects to show UI objects obstructed by other UI Objects, delete a selected UI Page Object, and editing the currently selected page, by working with a Edit Page Dialog Box.

[0135] Object styles panel **307b** of layer inspector **307** displays all styles on the Applications Page and permits operations to be performed on objects, and is similar to panel **307a**. Examples of operations provided by object style panel **307b** include, but are not limited to: importing from either a user's private object library or a public object library; inserting a new object style, which can be inherited from a currently selected object, or from a previously defined style object; and editing a currently selected object style by working with an Edit Style Dialog Box.

[0136] Style attributes can be assigned many attributes, including the look, and behavior of any object that inherits these objects. In addition, List Layout Styles can be created or changed as required. A layout style can define a unbounded set of Complex List Layouts, including but not limited to: the number of lines per item in a list, the number of text and image elements and their location for each line for each item in the last, the color and font for each text element, and the vertical and horizontal offset for each image and text element.

[0137] Alerts Panel **307c** provides a way of providing alert pages, which can have many of the attributes of Application Pages, but they are only activated through an Event such as a user interaction, a network event, a timer event, or a system variable setting, and will be superimposed onto whatever is currently being displayed. Alert Pages all have transparent backgrounds, and they function as a template overlay, and can also have dynamic binding to real time content.

[0138] Resource inspector **309** is the primary panel for interactively working with UI objects that have been placed on the Canvas **305**. When a UI object is selected on Canvas 305, a user of authoring platform **110** may associate properties of the selected object by entering or selecting from resource inspector **309**. In one embodiment, resource inspector **309** includes five tab

selections: Setting Tab **309a**, Events Tab **309b**, Animation Tab **309c**, Color Tab **309d** which includes a color palette for selecting object colors, and Bindings Tab **309e**.

[0139] Settings Tab **309a** provides a dialog box for the basic configuration of the selected object including, but not limited to, name, size, location, navigation and visual settings. Depending upon the type of object, numerous other attributes could be settable. As an example, the Setting Tab for a Text Field may include dialog boxes to define the text field string, define the object style, set the font name, size and effects, set an object name, frame style, frame width, text attributes (text field, password field, numeric field, phone number, SMS number, URL request).

[0140] As an example of Setting Tab **309a**, FIG. 3B shows various selections including, but not limited to, setting **309a1** for the web page name, setting **309a2** for the page size, including selections for specific devices **130**, setting **309a3** indicating the width and height of the object, and setting **309a4** to select whether background audio is present and to select an audio file.

[0141] Figure 3C illustrates an embodiment of the Events Tab **309b**, which includes all end user interactions and time based operations. The embodiment of Events Tab **309b** in FIG. 3C includes, for example and without limitation, an Events and Services **309b1**, Advanced Interactive Settings **309b2**, Mouse State **309b3**, Object Selected Audio Setting **309b4**, and Work with Child Objects and Mouse Overs button **309b5**.

[0142] Events and Services **309b1** lists events and services that may be applied to the selected objects. These include, but are not limited to, going to external web pages or other Applications pages, either as a new page or by launching a new window, executing an Application or JavaScript method, pausing or exiting, placing a phone call or SMS message, with or without single or multiple Player download, show launch strip, or go back to previous page. Examples of events and services include, but are not limited to those listed in Table II

Goto External Web Page replacing Current Frame	ChoiceObject : Remove Icon from Launch	
	Strip	
Goto External Web Page Launched in a New	Goto a specific Internal Web Page with	
Window	Alert. "Backend Synchronization"	
Goto a specific Internal Web Page	Goto Widget Object	
Goto the next Internal Web Page	Generate Alert. "With a Fire Event"	
Goto External Web Page replacing the Top Frame	Send SMS Message from Linked Text	
	Field	
Execute JavaScript Method	Toggle Alert. "Display OnFocus, Hide	
	OffFocus"	
Pause/Resume Page Timeout	Execute an Application with Alert. "With	
	a Fire Event"	
Execute an Application	Goto Logical First Page	
Goto a specific Internal Web Page with setting	Generate Alert with Backend	
starting slide	Synchronization	
Exit Application	Send SMS Message with Share (Player	
	Download)	
Exit Player	Place PhoneCall from linked Text Field	
	with Share (Player Download)	
Place PhoneCall from linked Text Field	Send IM Alert from linked Text Field or	
	Text Area	
Text Field/Area: Send String on FIRE	Set and Goto Starting Page	
ChoiceObject : Add Icon to Launch Strip	Populate Image	
Text Field/Area: Send String on FIRE or Numeric	Preferred Launch Strip	
Keys		

## Table II. Events and Services

[0143] Advanced Interactive Settings **309b2** include Scroll Activation Enabled, Timeline Entry Suppressed, Enable Server Listener, Submit Form, Toggle Children on FIRE, and Hide Non-related Children, Mouse State **309b3** selections are Selected or Fire. When Mouse State Selected is chosen, Object Selected Audio Setting **309b4** of Inactive, Play Once, Loop, and other responses are presented. When Mouse State Fire is chosen, Object Selected Audio Setting **309b4** is replaced with FIRE Audi Setting, with appropriate choices presented.

[0144] When Work with Child Objects and Mouse Overs button **309b5** is selected, a Child Object Mode box pops up, allowing a user to create a child object with shortcut to Menu bar **301** actions that may be used define child objects.

[0145] Figure 3D illustrates one embodiment of an Animation Tab **309c**, which includes all animations and timelines. The Color Tab includes all the possible color attributes, which may vary significantly by object type.

[0146] Animation Tab **309c** includes settings involved in animation and timelines that may be associated with objects. One embodiment of Animation Tab **309c** is shown, without limitation, in Figure 3D, and is described, in Rempell ("Rempell").

[0147] A Color Tab **309d** includes a color palette for selecting object colors.

[0148] Bindings Tab **309e** is where web component operations are defined and dynamic binding settings are assigned. Thus, for example, a UI object is selected from canvas **305**, and a web component may be selected and configured from the bindings tab. When the user's work is saved, binding information is associated with the UI object that will appear on screen **137**.

[0149] Figure 3E illustrates one embodiment of Bindings Tab and includes, without limitation, the following portions: Web Component and Web Services Operations **309e1**, Attributes Exposed list **309e2**, panel **309e3** which includes dynamic binding of server-side data base values to attributes for the selected object, Default Attribute Value **309e4**, Database Name **309e5**, Table Name **309e6**, Field Name **309e7**, Channel Name **309e8**, Channel Feed **309e9**, Operation **309e10**, Select Link **309e11**, and Link Set checkbox **309e12**.

[0150] Web Component and Web Services Operations **309e1** includes web components that may be added, edited or removed from a selected object. Since multiple web components can be added to the same object, any combination of mash-ups of 3rd party web services is possible. When the "Add" button of Web Component and Web Services Operations **309e1** is selected, a pop-up menu **319**, as shown in Figure 3F, appears on publisher interface **300**. Pop-up menu **319** includes, but is not limited to, the options of: Select a Web Component **319a**; Select Results Page **319b**; Activation Options **319c**; Generate UI Objects **319d**; and Share Web Component **319e**.

[0151] The Select a Web Component **319a** portion presents a list of web components. As discussed herein, the web components are registered and are obtained from web component registry **220**.

[0152] Select Results Page **319b** is used to have the input and output on different pages - that is, when the Results page is different from Input page. The default selected results page is either the current page, or, if there are both inputs and outputs, it will be set provisionally to the next page in the current page order, if one exists.

[0153] Activation Options **319c** include, if there are no Input UI Objects, a choice to either "Preload" the web component, similar to how dynamic binding, or have the web component executed when the "Results" page is viewed by the consumer.

[0154] Generate UI Objects **319c**, if selected, will automatically generate the UI objects. If not selected, then the author will bind the Web Component Inputs and Results to previously created UI Objects.

[0155] Share Web Component **319e** is available and will become selected under the following conditions: 1) Web Component is Selected which already has been used by the current Application; or 2) the current Input page is also a "Result" page for that Web component. This permits the user of device **130**, after viewing the results, to extend the Web Component allowing the user to make additional queries against the same Web Component. Examples of this include, but are not limited to, interactive panning and zooming for a Mapping Application, or additional and or refined searches for a Search Application.

[0156] Dynamic Binding permits the binding of real time data, that could either reside in a 3<sup>rd</sup> party server-side data base, or in the database maintained by Feed Collector **1010** for aggregating live RSS feeds, as described subsequently with reference to FIG. 10.

[0157] Referring again to FIG. 3E, Attributes Exposed list **309e2** are the attributes available for the selected object that can be defined in real time through dynamic binding.

[0158] Panel **309e3** exposes all the fields and tables associated with registered server-side data bases. In one embodiment, the user would select an attribute from the "Attributes Exposed List" and then select a data base, table and field to define the real time binding process. The final step is to define the record. If the Feed Collector data base is selected, for example, then the RSS "Channel Name" and the "Channel Feed" drop down menus will be available for symbolically selected the record. For other data bases the RSS "Channel Name" and the "Channel Feed" drop down menus are replaced by a "Record ID" text field.

[0159] Default Attribute Value **309e4** indicates the currently defined value for the selected attribute. It will be overridden in real time based on the dynamic linkage setting.

[0160] Database Name **309e5** indicates which server side data base is currently selected.

[0161] Table Name **309e6** indicates which table of the server side data base is currently selected.

[0162] Field Name **309e7**, indicates which field form the selected table of the server side data base is currently selected.

30

[0163] Channel Name **309e8** indicates a list of all the RSS feeds currently supported by the Feed Collector. This may be replaced by "Record ID" if a data base other than the Feed Collector **1010** is selected.

[0164] Channel Feed **309e9** indicates the particular RSS feed for the selected RSS Channel. Feed Collector **1010** may maintain multiple feeds for each RSS channel.

[0165] Operation **309e10**, as a default operation, replaces the default attribute value with the real time value. In other embodiments this operation could be append, add, subtract, multiply or divide.

[0166] Select Link **309e11** a button that, when pressed, creates the dynamic binding. Touching the "Select Link" will cause the current data base selections to begin the blink is some manner, and the "Select Link" will change to "Create Link". The user could still change the data base and attribute choices. Touching the "Create Link" will set the "Link Set" checkbox and the "Create Link" will be replaced by "Delete Link" if the user wishes to subsequently remove the link. When the application is saved, the current active links are used to create the SPDL.

[0167] Link Set checkbox **309e12** indicates that a link is currently active.

[0168] An example of the design of a display is shown in Figures 4A and 4B according the system 100, where FIG. 4A shows publisher interface 300 having a layout 410 on canvas 305, and FIG. 4B shows a device 130 having the resulting layout 420 on screen 137. Thus, for example, authoring platform 110 is used to design layout 410. Authoring platform 110 then generates an Application and a Player specific to device 130 of FIG. 4B. The Application and Player are thus used by device 130 to produce layout 420 on screen 137.

[0169] As illustrated in FIG. 4A, a user has placed the following on canvas **305** to generate layout **410**: text and background designs **411**, a first text input box **413**, a second text input box **415**, and a button **417**. As an example which is not meant to limit the scope of the present invention, layout **410** is screen prompts a user to enter a user name in box **413** and a password in box **415**, and enter the information by clicking on button **417**.

[0170] In one embodiment, all UI objects are initially rendered as Java objects on canvas**305**. When the Application is saved, the UI objects are transformed into the PDL, as described subsequently.

[0171] Thus, for example, layout **410** may be produced by the user of authoring platform **110** selecting and placing a first Text Field as box **413** then using the Resource Inspector **309** portion of interface **300** to define its attributes.

# DEVICE USER EXPERIENCE

[0172] Systems **100** and **200** provide the ability for a very large number of different types of user experiences. Some of these are a direct result of the ability of authoring platform **110** to bind UI objects to components of web services. The following description is illustrative of some of the many types of experiences of using a device **130** as part of system **100** or **200**.

[0173] Device **130** may have a one or more of a very powerful and broad set of extensible navigation objects, as well as object- and pointer-navigation options to make it easy to provide a small mobile device screen **137** with content and to navigate easily among page views, between Applications, or within objects in a single page view of an Application.

[0174] Navigation objects include various types of launch strips, various intelligent and userfriendly text fields and scrolling text boxes, powerful graphical complex lists, as well as Desktoplevel business forms. In fact, every type of object can be used for navigation by assigning a navigation event to it. The authoring tool offers a list of navigation object templates, which then can be modified in numerous ways.

# Launch Strips and Graphical List Templates Launch Strips

[0175] Launch strips may be designed by the user of authoring platform **110** with almost no restrictions. They can be stationary or appear on command from any edge of the device, their size, style, audio feedback, and animations can be freely defined to create highly compelling experiences.

[0176] Figure 5 shows a display **500** of launch strips which may be on displayed canvas **305** or on screen **137** of device **130** having the proper Player and Application. Display **501** includes a portal-type Launch Strip **501** and a channel-type Launch Strip **502**, either one of which may be included for navigating the Application.

[0177] Launch Strip 501 includes UI objects 501a, 501b, 501c, 501d, and 501e that that becomes visible from the left edge of the display, when requested. UI objects 501a, 501b, 501c, 501d, and 501e are each associated, through resource inspector 309 with navigational instructions, including but not limited to navigating to a different Applications Page, or

requesting web content. When the Applications Page, having been saved by authoring platform **110** and transferred to display **130**, is executed on device **130**, a user of the device may easily navigate the Application.

[0178] Launch Strip **502** includes UI objects **502b**, **502c**, **502d**, and **503e** that that becomes visible from the bottom of the display, when requested. UI objects **501a**, **501b**, **501c**, **501d**, and **501e** are each associated, through resource inspector **309** with navigational instructions, including but not limited to navigating to a different Applications Page, or requesting web content. Launch Strip **502** also includes UI objects **502a** and **503g**, which include the graphic of arrows, and which provide access to additional navigation objects (not shown) when selected by a user of device **130.** Launch strip **502** may also include sound effects for each channel when being selected, as well as popup bubble help.

[0179] Additional navigational features are illustrated in Figure 6A as a display of a Channel Selection List **601a**, in Figure 6B as a display of a Widget Selection List **601b**, and in Figure 6C as display of a Phone List **601c**. Lists **601a**, **601b**, and **601c** may be displayed on canvas **305** or on screen **137** of device **130** having the proper Player and Application. As illustrated, graphical lists **601a**, **601b**, and **601c** may contain items with many possible text and image elements. Each element can be defined at authoring time and/or populated dynamically through one or more Web Service **250** or API. Assignable Navigation Events. All objects, and/or all elements within an object, can be assigned navigation events that can be extended to registered web services or APIs. For example, a Rolodex-type of navigation event can dynamically set the starting slide of the targeted page view (or the starting view of a targeted Application).

[0180] In the embodiment of FIGS. 6A, 6B, and 6C, each list **601a**, **601b**, and **601c** has several individual entries that are each linked to specific actions. Thus Channel Selection List **601a** shows three objects, each dynamically linked to a web service (ESPN, SF 49ers, and Netflix) each providing a link to purchase or obtain items from the Internet. Widget Selection List **601b** includes several objects presenting different widgets for selecting. Phone List **601c** includes a list phone number objects of names that, when selected by a user of device **130** cause the number to be dialed Entries in Phone List **601c** may be generated automatically from either the user's contact list that is resident on the device, or though a dynamic link to any of user's chosen server-side facilities such as Microsoft Outlook, Google Mail, etc. In one embodiment, Phone List **601c** may be generated automatically using a web component assigned to the Application, which would automatically perform those functions.

33

[0181] In another embodiment, authoring platform **110** allows a navigation selection of objects with a Joy Stick and/or Cursor Keys in all 4 directions. When within a complex object the navigation system automatically adopts to the navigation needs for that object. For coordinate sensitive objects such as geographical information services (GIS) and location-based services (LBS) or virtual tours a soft cursor appears. For Lists, scrolling text areas and chats, Launch strips, and slide shows the navigation process permits intuitive selection of elements within the object. Scroll bars and elevators are optionally available for feedback. If the device has a pointing mechanism then scroll bars are active and simulate the desktop experience.

### Personalization and Temporal Adoption

[0182] System **100** and **200** permit for the personalization of device **130** by a variety of means. Specifically, what is displayed on screen **137** may depend on either adoption or customization. Adoption refers to the selection of choices, navigation options, etc. are based on user usage patterns. Temporal adoption permits the skins, choices, layouts, content. widgets, etc. to be further influenced by location (for example home, work or traveling) and time of day (including season and day of week). Customization refers to user selectable skins, choices, layouts, dynamic content, widgets, etc. that are available either through a customization on the phone or one that is on the desktop but dynamically linked to the user's other internet connected devices.

[0183] To support many personalization functions there must be a convenient method for maintaining, both within a user's session, and between sessions, memory about various user choices and events. Both utilizing a persistent storage mechanism on the device, or a database for user profiles on a server, may be employed.

[0184] Figure 7 shows a display 700 of a mash-up which may be on displayed canvas 305 or on screen 137 of device 130 having the proper Player and Application. Display 700 includes several object 701 that have been dynamically bound, including an indication of time 701a, an indication of unread text messages 701b, an RSS news feed 701c (including 2 "ESPN Top Stories" 701c1 and 701c2), components 701d from two Web Services - a weather report ("The Weather Channel"), and a traffic report 701e ("TRAFFIC.COM").

[0185] In assembling the information of display 700, device 130 is aware of the time and location of the device - in this example the display is for a workday when a user wakes. Device 130 has been customized so that on a work day morning the user wishes to receive the displayed information. Thus in the morning, any messages received overnight would be flagged, the user's

favorite RSS sports feeds would be visible, today's weather forecast would be available, and the current traffic conditions between the user's home and office would be graphically depicted. User personalization settings may be maintained as persistent storage on device **130** when appropriate, or in a user profile which is maintained and updated in real-time in a server-side data base.

#### PUSH CAPABLE SYSTEMS

[0186] In another embodiment system **100** or **200** is a push-capable system. As an example, of such systems, short codes may be applied to cereal boxes and beverage containers, and SMS text fields can be applied to promotional websites. In either case, a user of device **130** can text the short code or text field to an SMS server, which then serves the appropriate Application link back to device **130**.

[0187] Figure 8 is a schematic of an embodiment of a push enabled system 800. System 800 is generally similar to system 100 or 200. Device 130 is shown as part of a schematic of a push capable system 800 in Figure 8. System 800 includes a website system 801 hosting a website 801, a server 803 and a content server 805. System 801 is connected to servers 803 and/or 805 through the Internet. Server 803 is generally similar to server 120, servers 805 is generally similar to server 140.

[0188] In one embodiment, a user sets up a weekly SMS update from website system **801**. System **801** provides user information to server **803**, which is an SMS server, when an update is ready for delivery. Server **803** provides device **130** with an SMS indication that the subscribed information is available and queries the user to see if they wish to receive the update. Website **801** also provides content server **805** with the content of the update. When a user of device **130** responds to the SMS query, the response is provided to content server **805**, which provides device **130** with updates including the subscribed content.

[0189] In an alternative embodiment of system 800, server 803 broadcasts alerts to one or more devices 130, such as a logical group of devices. The user is notified in real-time of the pending alert, and can view and interact with the massage without interrupting the current Application.

[0190] Figure 9 is a schematic of an alternative embodiment of a push enabled system **900**. System **900** is generally similar to system **100**, **200**, or **800**. In system **900** a user requests information using an SMS code, which is delivered to device **130**. System **900** includes a promotional code **901**, a third-party server **903**, and content server **805**. Server **803** is connected to servers **803** and/or **805** through the Internet, and is generally similar to server **120**. Docket Number XPR.002WO0 35 [0191] A promotional code 901 is provided to a user of device 130, for example and without limitation, on print media, such as on a cereal box. The use of device 130 sends the code server 903. Server 903 then notifies server 805 to provide certain information to device 130. Server 805 then provides device 130 with the requested information.

# **DEVICE ROUTINES**

[0192] Device routines **114** may include, but are not limited to: an authoring tool SDK for custom code development including full set of Java APIs to make it easy to add extensions and functionality to mobile Applications and tie Applications to back-end databases through the content server **140**; an expanding set of web services **250** available through the authoring tool SDK; a web services interface to SOAP/XML enabled web services; and an RSS/Atom and RDF feed collector **1010** and content gateway **1130**.

## Authoring tool SDK for custom code development including full set of Java APIs

[0193] In one embodiment, authoring platform **110** SDK is compatible for working with various integrated development environments (IDE) and popular plug ins such as J2ME Polish. In one embodiment the SDK would be another plug in to these IDEs. A large and powerful set of APIs and interfaces are thus available through the SDK to permit the seamless extension of any Application to back end business logic, web services, etc. These interfaces and APIs may also support listeners and player-side object operations.

[0194] There is a large set of listeners that expose both player-side events and dynamically linked server side data base events. Some examples of player side events are: player-side time based event, a page entry event, player-side user interactions and player-side object status. Examples of server-side data base events are when a particular set of linked data base field values change, or some filed value exceeds a certain limit, etc.

[0195] A superset of all authoring tool functionality is available through APIs for layer-side object operations. These include, but are not limited to: page view level APIs for inserting, replacing, and or modifying any page object; Object Level APIs for modifying any attribute of existing objects, adding definitions to attributes, and adding, hiding or replacing any object.

# Authoring tool SDK available Web Services

[0196] The APIs permit, without limit, respond, with or without relying on back-end business logic, that is, logic that what an enterprise has developed for their business, to any player-side

event or server-side dynamically linked data-base, incorporating any open 3rd party web service(s) into the response.

### RSS/ATOM AND RDF FEED CONVERSION WEB SERVICE

[0197] Figure 10 is a schematic of one embodiment a system 1000 having a feed collector 1010. System 1000 is generally similar to system 100, 200, 800, or 900. Feed collector 1010 is a server side component of system 100 that collects RSS, ATOM and RDF format feeds from various sources 1001 and aggregates them into a database 1022 for use by the Applications built using authoring platform 110.

[0198] Feed collector **1010** is a standard XML DOM data extraction process, and includes Atom Populator Rule **1012**, RSS Populator Rule **1013**, RDF Populator Rule **1014**, and Custom Populator Rule **1016**, DOM XML Parsers **1011**, **1015**, and **1017**, Feed Processed Data Writer **1018**, Custom Rule Based Field Extraction **1019**, Rule-based Field Extraction **1020**, Channel Data Controller **1021**, and Database **1022**.

[0199] The feed collector is primarily driven by two sets of parameters: one is the database schema (written as SQL DDL) which defines the tables in the database, as well as parameters for each of the feeds to be examined. The other is the feed collection rules, written in XML, which can be used to customize the information that is extracted from the feeds. Each of the feeds is collected at intervals specified by the feed parameter set in the SQL DDL.

[0200] Feed collector **1010** accepts information from ATOM, RDF or RSS feed sources **1001.** Using a rules-based populator, any of these feeds can be logically parsed, with any type of data extraction methodology, either by using supplied rules, or by the author defining their own custom extraction rule. The rules are used by the parser to parse from the feed sources, and the custom rule base field extraction replaces the default rules and assembles the parsed information into the database

[0201] In particular, Atom Populator Rule **1012**, RSS Populator Rule **1013**, RDF Populator Rule **1014**, Custom Populator Rule **1016**, and DOM XML Parsers **1011**, **1015**, and **1017** are parse information from the feeds **1001**, and Feed Processed Data Writer **1018**, Custom Rule Based Field Extraction **1019**, Rule-based Field Extraction **1020**, and Channel Data Controller **1021**, supply the content of the feeds in Database **1022**, which is accessible through content server **140**.

[0202] Figure 11 is a schematic of an embodiment of a system **1100** having a Mobile Content Gateway **1130**. System **1100** is generally similar to system **100**, **200**, **800**, **900**, or **1000**. System **1100** includes an SDK **1131**, feed collector **1010**, database listener **1133**, transaction server **1134**, custom code **1135** generated from the SDK, Java APIs, Web Services **1137**, and PDL snippets compacted objects **1139**. System **1100** accepts input from Back End Java Code Developer **1120** and SOAP XML from Web Services **1110**, and provides dynamic content to server **140** and Players to devices **130**.

[0203] In one embodiment authoring platform **110** produces a Server-side PDL (SPDL) at authoring time. The SPDL resides in server **120** and provides a logical link between the Application's UI attributes and dynamic content in database **1022**. When a user of device **130** requests dynamic information, server **120** uses the SPDL to determine the link required to access the requested content.

[0204] In another embodiment Web Services **1137** interface directly with 3rd party Web Services **1110**, using SOAP, REST, JAVA, JavaScript, or any other interface for dynamically updating the attributes of the Application's UI objects.

#### XSP WEB PAGES AS A WEB SERVICE

[0205] In one embodiment, a PDL for a page is embedded within an HTML shell, forming one XSP page. The process of forming XSP includes compressing the description of the page and then embedding the page within an HTML shell.

[0206] In another embodiment, a PDL, which contains many individual page definitions, is split into separate library objects on the server, so that each page can to presented as a PDL as part of a Web Service.

[0207] Prior to compression the code has already been transformed so that there are no dependencies on the original programming language (Java), and The code and data have been reduced by 4 to 10 times.

[0208] Compression has two distinct phases. The first takes advantage of how the primitive representations had been assembled, while the second utilizes standard LZ encoding.

[0209] The final result is an overall reduction of 40 to 100 times the original size as represented by Java serialized objects.

38

[0210] One embodiment for compacting data that may be used is described in Rempell. In that patent the compressed data is described as being a database. The terminology used here is a PDL, that is the "internal database" of Rempell is equivalent to the PDL of the present Application.

[0211] In Rempell, a process for compacting a "database" (that is, generating a compact PDL) is described, wherein data objects, including but not limited to, multi media objects such as colors, fonts, images, sound clips, URLs, threads, and video, including multi level animation, transformation, and time line are compacted. As an extension to Rempell in all cases these objects are reduced and transformed to Boolean, integer and string arrays.

[0212] The compression technique involves storing data in the smallest arrays necessary to compactly store web page information. The technique also includes an advanced form of delta compression that reduces integers so that they can be stored in a single byte, a shigh water marks.

[0213] Thus, for example, the high water mark for different types of data comprising specific web site settings are stored in a header record as Boolean and integer variables and URL and color objects. Data that defines web page, paragraph, text button, and image style and text button, image and paragraph high watermark settings can be stored in one-dimensional arrays as Boolean, integer and string variables and URL, font, image or thread objects at. The URL, color, font, image and thread objects can also be created as required

[0214] Data that defines text button, image, paragraph, or other parent objects and paragraph line high watermark settings can be stored in two-dimensional arrays (by web page and by object number) as Boolean, integer, string, floating point variables and URLs. Again, the URL, color, font, image, audio clip, video clip, text area and thread objects can also be created as required. Data that defines a paragraph line and paragraph line segment high watermarks can be stored in three-dimensional arrays (by web page, by paragraph number, and by line number) as Boolean, integer or string variables. Again, the URL, color or font objects can be created as required. Data that defines a paragraph line segment can be stored into four-dimensional arrays (by web page, by paragraph number, by line number and by line number segment) as Boolean, integer or string variables or URL, color and font objects.

[0215] As a data field is added, changed or deleted, a determination is made at on whether a value for a given high watermark needs to be changed. If so, it is updated. As a specific method in the build engine is called, a determination is made on whether a feature flag needs to be set.

For example, if a particular JAVA method is called, which requires an instance of a certain JAVA Class to be executed by the run time engine, then that JAVA Class is flagged, as well as any supporting methods, variables and/or object definitions.

[0216] In one implementation, the header record, the style record, the web page record, and the object records, are carefully defined in a specific order, written in that order, and explicitly cast by object type when read by the run time engine. Exception handling can be implemented to recover from any errors. This helps assure that data integrity is maintained throughout the build and run time processes.

[0217] Also described in Rempell is the "run generation process." This is equivalent generating a Player in the present application. This process starts when the build process detects that the user is finished defining the web site (user has saved the web site and invokes the run generation process), and concludes with the actual uploading of all the necessary web site run time files to the user's server.

[0218] In one embodiment, the PDL includes a first record, a "Header" record, which contains can include the following information:

[0219] 1: A file format version number, used for upgrading database in future releases.

[0220] 2: The default screen resolution, in virtual pixels, for both the screen width and height. This is usually set to the web designer's screen resolution, unless overwritten by the user.

[0221] 3: Whether the Application is a web site.

[0222] 4: Virtual web page size settings. A calculation is performed by the build engine method, in order to calculate what the maximum web page length is, after reformatting all paragraphs on all internal web pages, based on the default screen resolution.

[0223] 5: Web page and styles high watermarks.

[0224] 6: The Websitename.

[0225] As new web pages or new objects are created by the user, or as text is added to or deleted from a paragraph, or as new styles are created or deleted, appropriate high watermarks are set, in order to show the current number of each of these entities. Thus, the values for the number of active web pages and the number of text button, image, paragraph or other styles are written as high watermarks in the header. The high watermarks for the number of text button,

image, paragraph or other objects that exist for each web page, the number of lines for each paragraph object, and the number of line segments for each paragraph line are written within the body of the PDL, and used as settings for each of the loops in the four-dimensional data structure. Because no structural limits are set on the number of web pages, objects per web page, styles, or paragraph size, these high watermarks greatly reduce the external database file size, and the time it takes for the run time engine to process the data stored in its database.

[0226] The settings for all paragraph, text button and image styles are then written as a style record based on their high watermark. This data includes Boolean and integer variables, and font and color objects, written as a one-dimensional array, based on the high watermark values for the number of styles that exist.

[0227] The body of the PDL is then written. All Boolean values are written inside a fourdimensional loop. The outside loop contains the Boolean values used to define web pages (i.e. a one-dimensional array definition) as well as the high watermarks for the number of text button, image, paragraph or other objects per web page, with the loop set at the high watermark which defines the number of existing web pages for this web site structure. The second level consists of three or more two dimensional loops with the loops set to the high watermarks defining the actual number of text button, image, and paragraph or other objects that appear on any given web page and contains the values used to define web page objects ((i.e. a two-dimensional array definition; web page number by object number). Included within the loop for paragraph objects are the high watermarks for the number of lines for each paragraph object. The third loop is set by the high watermark defining the actual number of paragraph lines that for all paragraphs on any web page and contains the values used to define paragraph lines (i.e. a three-dimensional array definition; web page number by object number by paragraph line.) Included within the loop for paragraph lines are the high watermarks for the number of line segments for each paragraph line. The inner most loop is set by the high watermarks defining the number of line segments per paragraph line and contains the values used to define paragraph line segments (i.e. a four-dimensional array definition; web page number by object number by paragraph line by paragraph line segment).

[0228] All integer values are written inside a four-dimensional loop. Their four loops are controlled by the same high watermark settings as used for the Boolean records, and they describe the same logical entities.

[0229] Multimedia objects are written inside a two-dimensional loop. They include URL, color, and font objects, and can include other types of objects. A URL object is the encoded form of a URL Address, used by a web browser or a JAVA method to access files and web addresses. Docket Number XPR.002WO0 41

All multimedia objects must be serialized before they can be written. This means that the objects are converted into a common external definition format that can be understood by the appropriate deserialization technique when they are read back in and cast into their original object structure. The outside loop contains web page related objects, and the inner loop contains image, text button, paragraph, etc. related URL, color, and font objects. The outer loop is defined by the web page high watermark and the inner loops by the high watermarks for the actual number of text button, image, paragraph or other objects on a web page.

[0230] String records are written inside a four-dimensional loop. The outer loop may be empty. The second loop can include the string values for text button objects, audio and video filenames, and audio and video channel names. The third loop contains values for paragraph line related data, and the innermost loop contains the values for paragraph line segment definitions. The string records are controlled by the same high watermarks as those used for Boolean and integer records. String records are stored utilizing an appropriate field delimiter technology. In one implementation, a UTF encoding technology that is supported by JAVA is utilized.

[0231] Single and double floating-point, and long integer records are written inside a twodimensional loop. The outer loop may be empty. The inner loop contains mathematical values required for certain animations and image processing algorithms. The single and double floatingpoint, and long integer records are controlled by the same high watermarks as those used for Boolean and integer records.

[0232] In one embodiment, a versionizing program analyzes the feature flags, and only those variable definitions, defined in the "Main" object class, relating to the object classes and methods that will be executed at run time, are extracted. All references to object classes that will be called at run time are extracted, creating the source code for the run engine "Main" object class that is ready for compilation.

[0233] All external image, video and audio files are resolved. The external references can be copied to designated directories, either on the user's local disk or file server. The file Pathnames can be changed to reflect these new locations. During the installation of the build tools, the necessary class libraries are either installed on the local system or made available on the server where the build tools can be optionally located. The necessary environmental variables are set to permit normal access to the required class libraries.

[0234] The customized run engine and a library of the referenced run time classes are compiled and converted into byte code. Finally, the run time engine for the web site is created.

Docket Number XPR.002WO0

42

The required set of class objects required at run time is flagged for inclusion into the CAB/JAR file.

[0235] Next, an HTML Shell File (HSF) is constructed. The first step of this process is to determine whether the dynamic web page and object resizing is desired by testing the Application setting. If the Application was a web page, and thus requiring dynamic web page and object resizing, virtual screen resolution settings are placed in an appropriate HTML compliant string. If the Application is a banner or other customized Application, the absolute values for the run time object (applet size) height and width are placed in an appropriate HTML compliant string as absolute width and height values.

[0236] An analysis is made for the background definition for the first internal web page. If a background pattern is defined, an appropriate HTML compliant string for setting the HTML "background" to the same background image is generated. If the first web page definition is a color instead, then the RGB values from those colors are converted to hexadecimal and an appropriate HTML compliant String is generated setting the "bgcolor" to the required hexadecimal value. This process synchronizes the web page background with the background that will be drawn by the web browser when it first interprets the HSF.

[0237] Thereafter, a JAVA method generates HTML and JavaScript compliant strings, that when executed by a web browser, generate additional sets of HTML and JavaScript compliant strings that are again executed by the web browser. More specifically, if the Application required dynamic web page and object resizing then JavaScript and HTML compliant strings are generated so that, when interpreted by the web browser at the time the HTML Shell File is initialized, the screen resolution sensing JAVA applet (SRS) will be executed. JavaScript code is generated in order to enable JavaScript to SRS applet communication. In one implementation, the code is generated by performing the following functions:

[0238] 1: Determine the current web browser type.

[0239] 2: Load the SRS from either a JAR or CAB File, based on web browser type.

[0240] 3: Enter a timing loop, interrogating when the SRS is loaded.

[0241] 4: When the SRS returns an "available" status, interrogate the SRS, which will return the current screen and window's actual height and width.

[0242] 5: Convert the virtual screen resolution settings into appropriate absolute screen width and height values.

[0243] Strings defining additional JavaScript code are generated that perform the following steps at the time the HSF is initialized by the web browser:

[0244] 1: Generate HTML compliant strings that set the run time engine's applet size to the appropriate values.

[0245] 2: Generate an HTML complaint string that contains a "param" definition for linking the run time engine to the PDL.

[0246] 3: Generate an HTML complaint string, dependent upon the type of web browser, which causes the current web browser to load either the JAR or the CAB File(s).

[0247] 4: Generate JavaScript Code compliant strings that create and dynamically write the applet size defining HTML strings utilizing the JavaScript "document.write" function. This dynamically created code causes the web browser to execute the run time engine, in the correctly sized window, from the correct JAR or CAB file, and linked to the external database.

[0248] The writing out the above-generated HTML and JavaScript compliant strings creates the HSF. The necessary security policy permissions are asserted, and a "Websitename".html file is created.

[0249] In one embodiment, the processes for creating the CAB and JAR Files is as follows. The image objects, if any, which were defined on the first internal web page are analyzed. If they are set to draw immediately upon the loading of the first web page, then they are flagged for compression and inclusion in the CAB and JAR Files. The feature flags are analyzed to determine which JAVA classes have been compiled. These class files are flagged for compression and inclusion in the library CAB and JAR Files. Strings that are BAT compliant definitions are created that will, when executed in DOS, create compressed CAB and JAR Files. These CAB and JAR Files contain the compressed versions of all necessary JAVA class files, image files, the "Websitename".class, customized run time engine file, and the "Websitename".dta database file. In one implementation of the invention, two BAT files are created. The first, when executed, will create a CAB/JAR file with the "Websitename".dta database file and the customized "main" run time engine, excluding all the image and button object animation, transformation, and image processing code. The second BAT file, when

44

executed, will create a CAB/JAR file with all the library of all the referenced image and button object animation, transformation, and image processing code.

[0250] The necessary security policy permissions for file creation are then asserted, and "Websitename".bat and "Websitename".bat and "Websitename".bat files are then executed under DOS, creating compressed "Websitename".cab and "Websitenamelib".cab files and compressed "Websitename" jar and "Websitenamelib" jar files. The HTML Shell File and the JAR and CAB files are then, either as an automatic process, or manually, uploaded to the user's web site. This completes the production of an XSP page that may be accessed through a web browser.

# DISPLAYING CONTENT ON A DEVICE

## Decompression Management

[0251] Authoring platform **110** uses compaction to transform the code and data in an intelligent way while preserving all of the original classes, methods and attributes. This requires both an intelligent server engine and client (handset) Player, both of which fully understand what the data means and how it will be used.

[0252] The compaction technology described above includes transformation algorithms that deconstruct the logic and data into their most primitive representations, and then reassembles them in a way that can be optimally digested by further compression processing. This reassembled set of primitive representations defines the PDL of authoring platform **110**.

[0253] Prior to compression the code has already been transformed so that there are no dependencies on the original programming language (Java). The data is then compressed by first taking advantage of how the primitive representations had been assembled, and then by utilizing standard LZ encoding. The final result is an overall reduction of 40 to 100 times the original size as represented by Java serialized objects.

[0254] The Player, when preparing a page view for execution, decompresses and then regenerate the original objects, but this time in compliance with the programming APIs of device **130**. Specifically, device **130** operates on compacted image pages, one at a time. The cache manager retrieves, decompresses, and reassembles the compacted page images into device objects, which are then interpreted by device **130** for display on screen **137**.

### **Response Director**

[0255] In one embodiment, system **100** includes a Response Director, which determines a user's handset, fetches the correct Application from different databases, and delivers a respective highly compressed Application in a PDL format over the air (OTA).

[0256] In one embodiment, the Response Director operates on a network connected computer to provide the correct Player to a given device based on the information the device sent to it. As an example, this may occur when a device user enters their phone number into some call-to-action web page. The response director is called and sends an SMS message to the device, which responds, beginning the recognition process.

[0257] Figure 12 illustrates one embodiment of a system **1200** that includes a response director **210**, a user agent database **1201**, an IP address database **1203**, and a file database **1205**. System **1200** is generally similar to system **100**, **200**, **800**, **900**, **1000**, or **1100**.

[0258] Databases **1201**, **1203**, and **1205** may reside on server **120**, **210**, or any computer system in communication with response director **210**. System **1200**, any mobile device can be serviced, and the most appropriate Application for the device will be delivered to the device, based on the characteristics of the device.

[0259] User agent database 1201 includes user agent information regarding individual devices 130 that are used to identify the operating system on the device. IP address database
1203 identifies the carrier/operator of each device 130. File database 1205 includes data files that may operate on each device 130.

[0260] The following is an illustrative example of the operation of response director **210**. First, a device **1300** generates an SMS message, which automatically sends an http:// stream that includes handset information and its phone number to response director **210**. Response director **210** then looks at a field in the http header (which includes the user agent and IP address) that identifies the web browser (i.e., the "User Agent"). The User Agent prompts a database lookup in user agent database **1201** which returns data including, but not limited to, make, model, attributes, MIDP 1.0 MIDP 2.0, WAP and distinguishes the same models from different countries. A lookup of the IP address in IP address **1203** identifies the carrier/operator.

[0261] File database **1205** contains data types, which may include as jad1, jad2, html, wml/wap2, or other data types, appropriate for each device **130**. A list of available Applications are returned to a decision tree, which then returns, to device **130**, the Application that is Docket Number XPR.002WO0 46

appropriate for the respective device. For each file type, there is an attributes list (e.g., streaming video, embedded video, streaming audio, etc.) to provide enough information to determine what to send to the handset.

[0262] Response director **210** generates or updates an html or jad file populating this text file with the necessary device and network dependent parameters, including the Application dependent parameters, and then generate, for example, a CAB or JAD file which contains the necessary Player for that device. For example, the jad file could contain the operator or device type or extended device-specific functions that the player would then become aware of.

[0263] If there is an Application that has a data type that device **130** cannot support, for example, video, response director **210** sends an alternative Application to the handset, for example one that has a slide show instead. If the device cannot support a slide show, an Application might have text and images and display a message that indicates it does not support video.

[0264] Another powerful feature of response director **210** is its exposed API from the decision tree that permits the overriding of the default output of the decision tree by solution providers. These solution providers are often licensees who want to further refine the fulfillment of Applications and Players to specific devices beyond what the default algorithms provide. Solution providers may be given a choice of Applications and then can decide to use the defaults or force other Applications.

[0265] Authoring platform **110** automatically scales Applications at publishing time to various form factors to reduce the amount of fragmentation among devices, and the Response Director serves the appropriately scaled version to the device. For example, a QVGA Application will automatically scale to the QCIF form factor. This is important because one of the most visible forms of fragmentation resides in the various form factors of wireless, and particularly mobile, devices, which range from 128x128, 176x208, 240x260, 220x220, and many other customized sizes in between.

[0266] Figure 13 is a schematic of an embodiment of a system **1300**. System **1300** is generally similar to system **1200**. System **1300** is an overview of the entire Player fulfillment process, starting with the generation of players during the player build process.

[0267] System 1300 includes response director 210, a device characteristics operator and local database 1301, a player profile database 1303 and a player build process 1305, which may be authoring platform 110.

[0268] As an example of system 1300, when response director 210 receives an SMS message from device 130, the response director identifies the device characteristics operator and locale from database 1301 and a Player URL from database 1303 and provides the appropriate Player to the device.

[0269] In another embodiment, Player **P** extend the power of response director **210** by adapting the Application to the resources and limitations of any particular device. Some of these areas of adaptation include the speed of the device's microprocessor, the presence of device resources such as cameras and touch screens. Another area of adaptation is directed to heap, record store and file system memory constraints. In one embodiment, the Player will automatically throttle down an animation to the frame rate that the device can handle so that the best possible user experience is preserved. Other extensions include device specific facilities such as location awareness, advanced touch screen interactions, push extensions, access to advanced phone facilities, and many others

#### Memory Management

[0270] In one embodiment, Player **P** includes a logical page virtual memory manager. This architecture requires no supporting hardware and works efficiently with constrained devices. All page view images, which could span multiple Applications, are placed in a table as highly compacted and compressed code. A typical page view will range from 500 bytes up to about 1,500 bytes. (See, for example, the Rempell patent) When rolled into the heap and instantiated this code increases to the more typical 50,000 up to 250,000 bytes. Additional alert pages may also be rolled into the heap and superimposed on the current page view. Any changes to any page currently downloaded are placed in a highly compact change vector for each page, and rolled out when the page is discarded. Note that whenever an Application is visited that had previously been placed in virtual memory the Server is interrogated to see if a more current version is available, and, if so, downloads it. This means that Application logic can be changed in real-time and the results immediately available to mobile devices.

[0271] To operate efficiently with the bandwidth constraints of mobile devices, authoring platform **110** may also utilize anticipatory streaming and multi-level caching. Anticipatory streaming includes multiple asynchronous threads and IO request queues. In this process, the

current Application is scanned to determine if there is content that is likely to be required in asyet untouched page views. Anticipatory streaming also looks for mapping Applications, where the user may zoom or pan next so that map content is retrieved prior to the user requesting it. For mapping applications, anticipatory streaming downloads a map whose size is greater than the map portal size on the device and centered within the portal. Any pan operation will anticipatory stream a section of the map to extend the view in the direction of the pan while, as a lower priority, bring down the next and prior zoom levels for this new geography. Zooming will always anticipatory stream the next zoom level up and down.

[0272] Multi-level caching determines the handset's heap through an API, and also looks at the record store to see how much memory is resident. This content is placed in record store and/or the file system, and may, if there is available heap, also place the content there as well. Multi-level caching permits the management of memory such that mobile systems best use limited memory resources. Multi-level caching is a memory management system with results similar to embedding, without the overhead of instantiating the content. In other words, with multi-level caching, handset users get an "embedded" performance without the embedded download. Note that when content is flagged as cacheable and is placed in persistent storage, a digital rights management (DRM) solution will be used.

[0273] One embodiment of each of the methods described herein is in the form of a computer program that executes on a processing system. Thus, as will be appreciated by those skilled in the art, embodiments of the present invention may be embodied as a method, an apparatus such as a special purpose apparatus, an apparatus such as a data processing system, or a carrier medium, e.g., a computer program product. The carrier medium carries one or more computer readable code segments for controlling a processing system to implement a method. Accordingly, aspects of the present invention may take the form of a method, an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects. Furthermore, the present invention may take the form of carrier medium (e.g., a computer program product on a computer-readable storage medium) carrying computer-readable program code segments embodied in the medium. Any suitable computer readable medium may be used including a magnetic storage device such as a diskette or a hard disk, or an optical storage device such as a CD-ROM.

[0274] It will be understood that the steps of methods discussed are performed in one embodiment by an appropriate processor (or processors) of a processing (i.e., computer) system executing instructions (code segments) stored in storage. It will also be understood that the

Docket Number XPR.002WO0

49

invention is not limited to any particular implementation or programming technique and that the invention may be implemented using any appropriate techniques for implementing the functionality described herein. The invention is not limited to any particular programming language or operating system. It should thus be appreciated that although the coding for programming devices has not be discussed in detail, the invention is not limited to a specific coding method. Furthermore, the invention is not limited to any one type of network architecture and method of encapsulation, and thus may be utilized in conjunction with one or a combination of other network architectures/protocols.

[0275] Reference throughout this specification to "one embodiment," "an embodiment," or "certain embodiments" means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, appearances of the phrases "in one embodiment," "in an embodiment," or "in certain embodiments" in various places throughout this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures or characteristics may be combined in any suitable manner, as would be apparent to one of ordinary skill in the art from this disclosure, in one or more embodiments.

[0276] Throughout this specification, the term "comprising" shall be synonymous with "including," "containing," or "characterized by," is inclusive or open-ended and does not exclude additional, unrecited elements or method steps. "Comprising" is a term of art which means that the named elements are essential, but other elements may be added and still form a construct within the scope of the statement. "Comprising" leaves open for the inclusion of unspecified ingredients even in major amounts.

[0277] Similarly, it should be appreciated that in the above description of exemplary embodiments, various features of the invention are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure and aiding in the understanding of one or more of the various inventive aspects. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed invention requires more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed embodiment, and the invention may include any of the different combinations embodied herein. Thus, the following claims are hereby expressly incorporated into this Mode(s) for Carrying Out the Invention, with each claim standing on its own as a separate embodiment of this invention.

50

[0278] Thus, while there has been described what is believed to be the preferred embodiments of the invention, those skilled in the art will recognize that other and further modifications may be made thereto without departing from the spirit of the invention, and it is intended to claim all such changes and modifications as fall within the scope of the invention. For example, any formulas given above are merely representative of procedures that may be used. Functionality may be added or deleted from the block diagrams and operations may be interchanged among functional blocks. Steps may be added or deleted to methods described within the scope of the present invention. We Claim:

1. A system for generating code to provide content on a display of a platform, said system comprising:

a database of web services obtainable over a network;

an authoring tool configured to:

define an object for presentation on the display,

select a component of a web service included in said database,

associate said object with said selected component, and

produce code that, when executed on the platform, provides said selected component on the display of the platform.

2. The system of Claim 1, where said database includes definitions of input and/or output related to said web service.

3. The system of Claim 1, where said component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

4. The system of Claim 1, where said object is an input field for a chat.

5. The system of Claim 1, where said object is an input field for a web service.

6. The system of Claim 1, where said object is an input field usable to obtain said component, where said input field includes a text field, a scrolling text box, a check box, a drop down-menu, a list menu, or a submit button.

7. The system of Claim 1, where said component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.

8. The system of Claim 1, where said authoring tool is further configured to:

define a phone field or list; and

generate code that, when executed on the platform, allows a user to supply a phone number to said phone field or list.

 The system of Claim 1, where said authoring tool is further configured to: define a SMS field or list; and

generate code that, when executed on the platform, allows a user to supply an SMS address to said SMS field or list.

10. The system of Claim 1, where said code includes two or more codes, where one of said two or more codes is platform specific, and where one of said two or more codes in platform independent.

11. The system of Claim 1, where said code is provided over said network.

12. A method for displaying content on a platform utilizing a database of web services obtainable over a network, said method comprising:

defining an object for presentation on the display; selecting a component of a web service included in said database; associating said object with said selected component; and producing code that, when executed on the platform, provides said selected component on the display of the platform.

13. The method of Claim 12, where said database includes definitions of input and/or output related to said web service.

14. The method of Claim 12, where said component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

15. The method of Claim 12, where said object is an input field for a chat.

16. The method of Claim 12, where said object is an input field for a web service.

17. The method of Claim 12, where said object is an input field usable to obtain said component, where said input field includes a text field, a scrolling text box, a check box, a drop down-menu, a list menu, or a submit button.

18. The method of Claim 12, where said component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.

19. The method of Claim 12, further comprising:

defining a phone field or list; and

generating code that, when executed on the platform, allows a user to supply a phone number to said phone field or list.

20. The method of Claim 12, further comprising:

defining a SMS field or list; and

generating code that, when executed on the platform, allows a user to supply an SMS address to said SMS field or list.

21. The method of Claim 12, where said code includes two or more codes, where one of said two or more codes is platform specific, and where one of said two or more codes in platform independent.

22. The method of Claim 12, further comprising:

providing said code over said network.

 A method for providing information to a platform on a network, said method comprising: accepting a first code over the network, where said first code is platformdependent;

providing a second code over the network, where said second code is platformindependent; and

executing said first code and said second code on the platform to provide web components obtained over the network.

24. The method of Claim 23, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

25. The method of Claim 23, where said component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.

26. The method of Claim 23, where said first code and said second code are generated using an authoring tool.

27. The method of Claim 23, where said first code is a Player.

28. The method of Claim 23, where said second code is an Application.

## SYSTEMS AND METHODS FOR PROGRAMMING MOBILE DEVICES

# ABSTRACT

Embodiments of a system and method are described for generating and distributing programming to mobile devices over a network. Devices are provided with Players specific to each device and Applications that are device independent. Embodiments include a full-featured WYSIWYG authoring environment, including the ability to bind web components to objects.


FIG. 1A

2/18 +100 110 Authoring Platform Α, Β <u>N</u> Α, Β A, B <u>120b</u> Server 120a Server N A-N, B-N -A-1,B-1 A-2, B-2 <u>130-1</u> Device 1 <u>130-N</u> Device N <u>130-2</u> Device 2 C-2, R-2 C-N, R-N C-1, R-1 N C, R Content <u>140</u> Server

FIG. 1B



FIG. 2A



5/18





IPR2021-01146 Page 00078



FIG. 3C

+

9c							
Settin	as [	Events	🖉 Ani	mation	Color	Bind	lings
Activate Timeline	e Oek Book	bject Entry ay (Sec)	Timeli Directi None	ne Specific on Moven	ations rent Dur 12	ation(Sec PI.o SI	) Fram 110
ene	ند. ا	unt (	s hie Ahi	nete Entra	aa <b>f Baaa</b> a Amineerstier	n Anstin T	3
Inactiv	0 0				~~~~~~~~~~~~	Entry A	
•							
St	×x	Object An	limatio	n Specifical	lions		*
	.0 .0	Scroll Lei	ft	Fade			1 <b>A</b> 2
2 3	.2 .3	Scroll Rig	iht	Fade In Fade Out	3	.2 .3	3
4	.4	Multi-Poir	nt.		4	4	5
5 6	с. 6.	Attach	sor		0	.5 .6	10 7
7	.7	Deposit			7	.7	8
8 9	.ŏ .9	Carom N	ne		8	.8 1.9	19
	I	Carom NI		1	110		11
	Р	athname fo	or this (	Objecťs Ani	imation A	udio Tra	:k
Inactiv	e	<b>.</b>				Main A	uudio Fi
Animat	ion C	ycles Cu: ]	stom Zo	xom % Avo	id Carso 🔽	r Damp	ien Anli E
	0	bject Exit 1	(imelin)	e Specifical	lions		
	· Dek	ny (Sec)	Directi None	on Moven	nent Dur 2	ation(Sec	) Fram [10]
Spe	cifica	ations for t	his Obj	ect's Exit A	nimation	Audio Tra	3CK
Inactiv	6						udio Fil

FIG. 3D



FIG. 3E

10/18





# FIG. 3F

IPR2021-01146 Page 00082



+





FIG. 7

14/18 800 . <u>801</u> Website System <u>803</u> <u>130</u> Device SMS Server <u>805</u> **Content Server** FIG. 8 900 ~ <u>901</u> Promo Code <u>805</u> <u>130</u> Device **Content Server** <u>903</u> **3rd Party Server** 

15/18 1000 ~ 1001 -**Feed Sources** Mobile Content Gateway Source 2 Source 1 Source A 1010 -1011 Atom/RSS/RDF 1012 Feed Collector DOM XM 1013 Perset Ø, 1014. -1016 ATOM 888 RDF Custom Populator Populator Populator Populator 80.15 1 Rule 🎆 Rule Ruis 1015-XML Dociment Tree -1017 DOM XM DOM XM Parses Perser Player requests Ø, (a) **Content Server** for Feed Data HTTP:80 POST -1018 130-~1019 Ø Ø Q Custom Rule Food Rule-based Based Field Field Processed -1020 Extraction Data Writer Extraction W U Ľ 日本日 -1021 ..... Database Update ٢ Channel Data. **DEDUCTION** Request by Channet, tem, Field Controller 13 Dynamic -1022 140 Content Server Client

FIG. 10

16/18







FIG. 13

PTO-1390 (Rev. 09-08) Approved for use through 3/31/2010. OMB 0651-0021 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

	ATTORNEY'S DOCKET NUMBER XPR.002US0				
	OFFICE (DO/EO/08) ON UNDER 35 U.S.C. 371	U.S. APPLICATION NO. (If known, see 37 CFR 1.5) Not Yet Assigned			
INTERNATIONAL APPLICATION NO. PCT/US09/039695	INTERNATIONAL FILING DATE April 6, 2009	PRIORITY DATE CLAIMED			
		, 1917, 2000			
APPLICANT(S) FOR DO/EO/US					
Express Mobile, Inc. Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:					
1. This is a <b>FIRST</b> submission of items c	oncerning a submission under 35 U.S.C. 37	1.			
2 This is a SECOND or SUBSEQUENT	submission of items concerning a submissio	n under 35 U.S.C. 371			
$3 \qquad \qquad$	tional examination procedures (35 U.S.C. 37	(1(f)) The submission must include items			
(5), (6), (9) and (21) indicated below.	ional examination procedures (55 0.0.0, 57				
4. The US has been elected (Article 31).					
5. A copy of the International Applicatio	on as filed (35 U.S.C. 371(c)(2))				
a. 🔄 is attached hereto (require	d only if not communicated by the Internation	nal Bureau).			
b. has been communicated b	y the International Bureau.				
c. 🔽 is not required, as the app	lication was filed in the United States Receiv	ing Office (RO/US).			
6. An English language translation of th	ne International Application as filed (35 U.S.	C. 371(c)(2)).			
a. is attached hereto.					
b. L has been previously subm	itted under 35 U.S.C. 154(d)(4).				
7. Amendments to the claims of the Int	ernational Application under PCT Article 19	(35 U.S.C. 371(c)(3))			
a. 🔲 are attached hereto (requ	ired only if not communicated by the Interna	itional Bureau).			
b. L have been communicated	d by the International Bureau.				
c. L have not been made; how	vever, the time limit for making such amendr	nents has NOT expired.			
d. 🖍 have not been made and	will not be made.				
8. An English language translation of t	he amendments to the claims under PCT Ar	ticle 19 (35 U.S.C. 371(c)(3)).			
9.  An oath or declaration of the invento	r(s) (35 U.S.C. 371(c)(4)).				
10. An English language translation of the Article 36 (35 U.S.C. 371(c)(5)).	ne annexes of the International Preliminary E	Examination Report under PCT			
Items 11 to 20 below concern document(	s) or information included:				
11. An Information Disclosure Statemen	t under 37 CFR 1.97 and 1.98.				
12. An assignment document for record	ing. A separate cover sheet in compliance w	ith 37 CFR 3.28 and 3.31 is included.			
13. A preliminary amendment.					
14. An Application Data Sheet under 37	CFR 1.76.				
15. A substitute specification.					
16. A power of attorney and/or change of	of address letter.				
17. A computer-readable form of the sec	quence listing in accordance with PCT Rule	13 <i>ter</i> .3 and 37 CFR 1.821- 1.825.			
18. A second copy of the published Inter	rnational Application under 35 U.S.C. 154(d)	(4).			
19. A second copy of the English langua	age translation of the international application	n under 35 U.S.C. 154(d)(4).			
L This collection of information is required by 37 CFR 1.41 USPTO to pro cess) an application . Confidentiality is dou	4 and 1.491-1.492. The information is required to verned by 35 U.S.C. 122 and 37 CFR 1.11 and	obtain or retain a benefit by the public, which is to file (and by th 1.14. This collection is estimated to take 15 minutes to complet			

This collection of information is required by 37 CFR 1.414 and 1.491-1.492. The information is required to obtain or retain a benefit by the public, which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 15 minutes to complete, including gathering information, preparing, and submitting the completed form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEE S OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450. Page 1 of 3

PTO-1390 (Rev. 09-08) Approved for use through 3/31/2010. OMB 0651-0021 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE to a collection of information unless it displays a valid OMB control number.

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE der the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

U.S. APPLICAT Not Yet Assig	TION NO. (if known Ined	, see 37 CFR 1.5	) INTERNATIONAL AI PCT/US09/039695	PPLICATION NO.	ATTORNEY'S DOC XPR.002US0	KET NUMBER
20. Other	items or informati	on:				
The foll	lowing fees have b	een submitted			CALCULATIONS	PTO USE ONLY
21. 🔲 Basi	ic national fee (37	CFR 1.492(a))		\$330	<sup>\$</sup> 330	
22. 🔲 Exan	nination fee (37 Cl	FR 1.492(c))				
If the written opin by IPEA/ All other situatio	nion prepared by IS /US indicates all cl ns	SA/US or the inte aims satisfy prov	ernational preliminary examina visions of PCT Article 33(1)-(4)	tion report prepared <b>\$0</b> <b>\$220</b>	\$220	
23 🔲 Sea	rch fee (37 CER 1	492(h))		•		
If the written opin	nion of the ISA/US	or the Internatio	nal preliminary examination re	port prepared by	400	
Search fee (37 C	FR 1.445(a)(2)) h	as been paid on	the international application to	the USPTO as an	\$ <sup>430</sup>	
International Sea	onal Searching Au arch Report prepar	thority ed by an ISA oth	ner than the US and provided t	o the Office or		
previously All other situatior	y communicated to	o the US by the I	В	\$430 \$540		
		2 and 23 =			980	
Additional fe	e for specification	and drawings file	ed in paper over 100 sheets (e	excluding sequence		
program	listing in an electr	onic medium) (3	or (e) in an electronic medium 7 CFR 1.492(j)).	or computer		
The fee is \$2	270 for each additi	onal 50 sheets c	of paper or fraction thereof.	1		
Total Sheets	Extra Sheets	Number of each thereof (round	n additional 50 or fraction I up to a whole number)	RATE		
- 100 =	/50 =			× \$270	\$ O	
Surcharge of <b>\$13</b> after the date of a	30.00 for furnishing commencement of	g any of the sear f the national sta	ch fee, examination fee, or the ge (37 CFR 1.492(h)).	oath or declaration	\$ O	
CLAIMS	NUMB	ER FILED	NUMBER EXTRA	RATE	\$	
Total claims	28	- 20 =	8	x \$ <b>52</b>	\$ 416	
Independent clair	ms 3	- 3 =	0	x \$220	\$ O	
MULTIPLE DEPI	ENDENT CLAIM(S	6) (if applicable)		+ \$390	\$ O	
			TOTAL OF ABOVE	CALCULATIONS =	\$1396	
Applicant cla	aims small entity st	tatus. See 37 CF	R 1.27. Fees above are redu	ced by ½.		
				SUBTOTAL =	\$ 698	
Processing fee or claimed priority d	Processing fee of <b>\$130.00</b> for furnishing the English translation later than 30 months from the earliest claimed priority date (37 CFR 1.492(i)). +				\$	
TOTAL NATIONAL FEE =				\$		
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31).         \$40.00 per property				\$ 40.00		
			TOTAL F	EES ENCLOSED =	\$ 738.00	
					Amount to be refunded:	\$
					Amount to be charged	\$ 738.00

а. 🗌	A check in the amount of \$	to cover the above fees is encl	osed.
b. 🗖	Please charge my Deposit Account No.	_ in the amount of \$	to cover the above fees.
c. 🗌	The Commissioner is hereby authorized to charge an Account No.	y additional fees which may be rea	quired, or credit any overpayment to Deposit
d. 🗖	Fees are to be charged to a credit card. <b>WARNING:</b> <b>be included on this form.</b> Provide credit card inform to the USPTO. However, when paying the basic nati	Information on this form may beco nation and authorization on PTO-20 onal fee, the PTO-2038 may NOT	me public. <b>Credit card information should not</b> 038. The PTO-2038 should only be mailed or faxed be faxed to the USPTO.
	<b>ADVISORY</b> : If filing by EFS-Web, do <b>NOT</b> attach t advised that this is <b>not</b> recommended and by doing information, it is recommended paying fees online is	he PTO-2038 form as a PDF along g so your <b>credit card information</b> by using the electronic payment m	g with your EFS-Web submission. Please be may be displayed via PAIR. To protect your ethod.
NOTE: ' and gra	Where an appropriate time limit under 37 CFR 1.49 Inted to restore the International Application to pen	5 has not been met, a petition to ding status.	revive (37 CFR 1.137(a) or (b)) must be filed
SEND A	ALL CORRESPONDENCE TO:	/Steven	R. Vosen/
Ste	ven B. Vosen	SIGNA	TURE
156	3 Solano Ave.	Steven	R. Vosen
#20	6	NAME	
Ber	keley, CA 94707	45,186	
		REGIS	TRATION NUMBER

# **Privacy Act Statement**

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

- 1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
- 2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
- 3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
- 4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
- 5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
- 6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
- 7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
- 8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
- 9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Patent Application Fee Transmittal					
Application Number:					
Filing Date:					
Title of Invention:	SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBIL DEVICES			ON ON MOBILE	
First Named Inventor/Applicant Name:	Steven H Rempell				
Filer:	Steven R Vosen				
Attorney Docket Number: XPR.002US0					
Filed as Small Entity					
U.S. National Stage under 35 USC 371 Filing	Fees	;			
Description		Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:					
Basic National Stage Fee		2631	1	165	165
Natl Stage Search Fee - Report provided		2642	1	215	215
Natl Stage Exam Fee - all other cases		2633	1	110	110
Pages:					
Claims:					
Claims in excess of 20		2615	8	26	208
Miscellaneous-Filing:	Miscellaneous-Filing:				
Petition:					

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
	Tot	al in USD	) (\$)	698

Electronic Acknowledgement Receipt				
EFS ID:	8525053			
Application Number:	12936395			
International Application Number:	PCT/US09/39695			
Confirmation Number:	2369			
Title of Invention:	SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES			
First Named Inventor/Applicant Name:	Steven H Rempell			
Customer Number:	40280			
Filer:	Steven R Vosen			
Filer Authorized By:				
Attorney Docket Number:	XPR.002US0			
Receipt Date:	04-OCT-2010			
Filing Date:				
Time Stamp:	20:34:37			
Application Type:	U.S. National Stage under 35 USC 371			

# Payment information:

Submitted with Payment			no			
File Listing:						
Document Number	Document Description		File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Assignee showing of ownership per 37	St	tatement_For_Assignment.	422434	no	2
	CFK 3.73(b).		рат	2876e4752bb28d9c133ec8aee1ee920538a a9a0d		
Warnings:						
Information:						

2		XPR002LISO PrelimAmt odf	113559	Vec	6		
2			1c68479bdecb9813537f61349f864a702d9 41b55	yes			
	Multip	oart Description/PDF files in	.zip description				
	Document De	Start	E	nd			
	Preliminary Am	Preliminary Amendment			1		
	Claims		2		5		
	Applicant Arguments/Remarks	Made in an Amendment	6		6		
Warnings:							
Information:		1	i				
3	Oath or Declaration filed	XPR002US0_DECandPOA_sign	1846031	no	6		
		ed2.pdf	b31e9a3ca8030b769f5a863e5af681a2b395 f59e				
Warnings:							
Information:		ł	1				
4	Application Data Sheet	XPR002US0_ADS.pdf	968697	no	5		
			3d57a8bf6bfee3388377d5086e48a9b48f45 5a5d	110			
Warnings:		·	·		-		
Information:	Information:						
5			550132	Ves	55		
5			aa5b1222722b85aa58ac91eda7d36c667e7 7b8c7	yes			
	Multip	art Description/PDF files in	zip description		-		
	Document De	scription	Start	E	nd		
	Specificat	ion	1		51		
	Claims	52		54			
	Abstrac	55		55			
Warnings:							
Information:		1	1		1		
6	Drawings-only black and white line drawings	XPR002WO0FIGS_BW.pdf	15355027 9b2fd94f419ba5d5bc9b9e6332924d58128 ~7750	no	18		
Warnings:			C//dU		1		
 Information:							

This Acknow	rledaement Receipt evidences receip	t on the noted date by the U	SPTO of the indicated	d document	:s,
		Total Files Size (in bytes)	: 19	861202	
Information	:				
Warnings:					
			fbdac71e1a779ff5747562c1868caa9062a4 92e6		
8	Fee Worksheet (PTO-875)	fee-info.pdf	36902	no	2
Information					
Warnings:					
,	Applications	c06583882736f9f6e328193aeb62c0ed202e 2246			
7	Documents submitted with 371	XPR002US0 Form 371 pdf	568420	no	4

## New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

## National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

## New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application. Г

for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

	STATEMENT UNDER 37	<u>CFR 3.73(b)</u>
Applicant/Patent Owner: Express	Mobile, Inc	
Application No./Patent No.:	File	ed/Issue Date:
Titled: SYSTEMS AND METH	ODS FOR PROGRAMMING MOBIL	E DEVICES
Express Mobile, Inc.	, <sub>a</sub> Corporation	
(Name of Assignee)	(Type of Assign	ee, e.g., corporation, partnership, university, government agency, etc.
states that it is:		
1. X the assignee of the entir	e right, title, and interest in;	
2. an assignee of less than (The extent (by percenta	the entire right, title, and interest in ige) of its ownership interest is	%); or
3. the assignee of an undiv	ided interest in the entirety of (a comple	te assignment from one of the joint inventors was made)
the patent application/patent identified	ed above, by virtue of either:	
A. X An assignment from the the United States Patent copy therefore is attache	inventor(s) of the patent application/pate t and Trademark Office at Reel 024341 ed.	ent identified above. The assignment was recorded in , Frame 0039, or for which a
OR		
B. A chain of title from the i	nventor(s), of the patent application/pate	ent identified above, to the current assignee as follows:
1. From:		То:
The document	t was recorded in the United States Pate	ent and Trademark Office at
Reel	, Frame	, or for which a copy thereof is attached.
2. From:		То:
The document	t was recorded in the United States Pate	ent and Trademark Office at
Reel	, Frame	, or for which a copy thereof is attached.
3. From:		То:
The document	t was recorded in the United States Pate	ent and Trademark Office at
Reel	, Frame	, or for which a copy thereof is attached.
Additional documents ir	າ the chain of title are listed on a suppler	nental sheet(s).
As required by 37 CFR 3.73 or concurrently is being, subr	(b)(1)(i), the documentary evidence of th nitted for recordation pursuant to 37 CFI	ne chain of title from the original owner to the assignee was, R 3.11.
[NOTE: A separate copy ( <i>i.e</i> accordance with 37 CFR Par	., a true copy of the original assignment t 3, to record the assignment in the reco	document(s)) must be submitted to Assignment Division in rds of the USPTO. <u>See</u> MPEP 302.08]
The undersigned (whose title is sup	plied below) is authorized to act on beha	If of the assignee.
/Steven R. Vosen/		September 29, 2010
Signature		Date
Steven R. Vosen (Reg No. 45,18	36)	Registered Patent Agent
Printed or Typed Name		Title
This collection of information is required by 37 process) an application. Confidentiality is gow gathering, preparing, and submitting the comp you require to complete this form and/or sugge Department of Commerce, P.O. Box 1450, Ale	CFR 3.73(b). The information is required to obtain erned by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. leted application form to the USPTO. Time will vary estions for reducing this burden, should be sent to th exandria, VA 22313-1450. DO NOT SEND FEES C	or retain a benefit by the public which is to file (and by the USPTO to This collection is estimated to take 12 minutes to complete, including / depending upon the individual case. Any comments on the amount of time he Chief Information Officer, U.S. Patent and Trademark Office, U.S. DR COMPLETED FORMS TO THIS ADDRESS. <b>SEND TO: Commissioner</b>

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2. IPR2021-01146 Page 00100

# **Privacy Act Statement**

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

- 1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
- 2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
- 3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
- 4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
- 5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
- 6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
- 7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
- 8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
- 9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

#### (19) World Intellectual Property Organization International Bureau

(43) International Publication Date

15 October 2009 (15.10.2009)



US

US

(51) International Patent Classification: *G06F 9/44* (2006.01) *G06Q 50/00* (2006.01) *G06F 17/00* (2006.01)

- (21) International Application Number: PCT/US2009/039695
- (22) International Filing Date:

6 April 2009 (06.04.2009)(25) Filing Language:English(26) Publication Language:English

- (30) Priority Data: 61/123,438 7 April 2008 (07.04.2008) 61/113,471 11 November 2008 (11.11.2008)
- 61/166,651
   3 April 2009 (03.04.2009)
   US

   (71)
   Applicant (for all designated States except US):
   EX 

   DEFEC
   MOPULE
   US
   12
   Descent US):
   EX
- PRESS MOBILE, INC. [US/US]; 12 Pamaron Way, Suite 1, Novato, CA 94947 (US).

#### (72) Inventors; and

(75) Inventors/Applicants (for US only): CHROBAK, David [US/US]; 132 Shadowood Drive, Pleasant Hill, CA (10) International Publication Number WO 2009/126591 A1

94523 (US). **BROWN, Ken** [US/US]; 2485 Church Avenue, San Martin, CA 95046 (US). **REMPELL, Steven, H.** [US/US]; 38 Washington Street, Novato, CA 94947 (US).

(74) Agent: VOSEN, Steven, R.; 1563 Solano Ave., #206, Berkeley, CA 94707 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE,

[Continued on next page]

#### (54) Title: SYSTEMS AND METHODS FOR PROGRAMMING MOBILE DEVICES

(57) Abstract: Embodiments of a system and method are described for generating and distributing programming to mobile devices over a network. Devices are provided with Players specific to each device and Applications that are device independent. Embodiments include a full-featured WYSIWYG authoring environment, including the ability to bind web components to objects.





\_\_\_\_

ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

- Published:
- with international search report (Art. 21(3))

before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

## SYSTEMS AND METHODS FOR PROGRAMMING MOBILE DEVICES

## TECHNICAL FIELD

[0001] The present invention generally relates to providing software for mobile devices, and more particularly to a method and system for authoring Applications for devices.

## BACKGROUND ART

[0002] Internet-connected mobile devices are becoming ever more popular. While these devices provide portability to the Internet, they generally do not have the capabilities of non-mobile devices including computing, input and output capabilities.

[0003] In addition, the mobility of the user while using such devices provides challenges and opportunities for the use of the Internet. Further, unlike non-mobile devices, there are a large number of types of devices and they tend to have a shorter lifetime in the marketplace. The programming of the myriad of mobile devices is a time-consuming and expensive proposition, thus limiting the ability of service providers to update the capabilities of mobile devices.

[0004] Thus there is a need in the art for a method and apparatus that permits for the efficient programming of mobile devices. Such a method and apparatus should be easy to use and provide output for a variety of devices.

# DISCLOSURE OF INVENTION

[0005] In certain embodiments, a system is provided to generate code to provide content on a display of a platform. The system includes a database of web services obtainable over a network and an authoring tool. The authoring tool is configured to define an object for presentation on the display, select a component of a web service included in said database, associate said object with said selected component, and produce code that, when executed on the platform, provides said selected component on the display of the platform.

[0006] In certain other embodiments, a method is provided for providing information to platforms on a network. The method includes accepting a first code over the network, where said first code is platform-dependent; providing a second code over the network, where said second code is platform-independent; and executing said first code and said second code on the platform to provide web components obtained over the network.

[0007] In certain embodiments, a method for displaying content on a platform utilizing a database of web services obtainable over a network is provided. The method includes: defining an object for presentation on the display; selecting a component of a web service included in said

#### WO 2009/126591

#### PCT/US2009/039695

database; associating said object with said selected component; and producing code that, when executed on the platform, provides said selected component on the display of the platform.

[0008] In one embodiment, one of the codes is a Player, which is a thin client architecture that operates in a language that manages resources efficiently, is extensible, supports a robust application model, and has no device specific dependencies. In another embodiment, Player **P** is light weight and extends the operating system and/or virtual machine of the device to: Manage all applications and application upgrades, and resolve device, operating system, VM and language fragmentation.

[0009] In another embodiment, one of the codes is an Application that is a device independent code that interpreted by the Player.

[0010] These features together with the various ancillary provisions and features which will become apparent to those skilled in the art from the following detailed description, are attained by the system and method of the present invention, preferred embodiments thereof being shown with reference to the accompanying drawings, by way of example only, wherein:

## BRIEF DESCRIPTION OF DRAWINGS

[0011] FIG. 1A is an illustrative schematic of one embodiment of a system including an authoring platform and a server for providing programming instructions to a device over a network;

[0012] FIG. 1B is schematic of an alternative embodiment system for providing programming instructions to device over a network;

[0013] FIG. 2A is a schematic of an embodiment of system illustrating the communications between different system components;

[0014] FIG.2B is a schematic of one embodiment of a device illustrating an embodiment of the programming generated by authoring platform;

[0015] FIGS. 3A and 3B illustrate one embodiment of a publisher interface as it appears, for example and without limitation, on a screen while executing an authoring tool;

[0016] FIG. 3C illustrates an embodiment of the Events Tab'

[0017] FIG. 3D illustrates one embodiment of an Animation Tab;

[0018] FIG. 3E illustrates one embodiment of Bindings Tab;

[0019] FIG. 3F illustrates one embodiment of a pop-up menu for adding web components;

[0020] FIG. 4A shows a publisher interface having a layout on a canvas; and FIG. 4B shows a device having the resulting layout on a device screen;

[0021] FIG. 5 shows a display of launch strips;

[0022] FIG. 6A is a display of a Channel Selection List;

[0023] FIG. 6B is a display of a Widget Selection List;

[0024] FIG. 6C is a display of a Phone List;

[0025] FIG. 7 shows a display of a mash-up;

[0026] FIG. 8 is a schematic of an embodiment of a push capable system;

[0027] FIG. 9 is a schematic of an alternative embodiment of a push capable system;

[0028] FIG. 10 is a schematic of one embodiment of a feed collector;

[0029] FIG. 11 is a schematic of an embodiment of a Mobile Content Gateway;

[0030] FIG. 12 is a schematic of one embodiment of a system that includes a response director, a user agent database, an IP address database, and a file database; and

[0031] FIG. 13 is a schematic of another embodiment of a system that includes a response director, a user agent database, an IP address database, and a file database.

[0032] Reference symbols are used in the Figures to indicate certain components, aspects or features shown therein, with reference symbols common to more than one Figure indicating like components, aspects or features shown therein.

## MODE(S) FOR CARRYING OUT THE INVENTION

[0033] Figure 1A is an illustrative schematic of one embodiment of a system 100 including an authoring platform 110 and a server 120 for providing programming instructions to a device 130 over a network N. In one embodiment, device 130 is a wireless device, and network N includes wireless communication to the device. Alternatively, system 100 may provide access over network N to other information, data, or content, such as obtainable as a web service over the

#### WO 2009/126591

#### PCT/US2009/039695

Internet. In general, a user of authoring platform **110** may produce programming instructions or files that may be transmitted over network **N** to operate device 130, including instructions or files that are sent to device **130** and/or server **120**. The result of the authoring process is also referred to herein, and without limitation, as publishing an Application.

[0034] Embodiments include one or more databases that store information related to one or more devices **130** and/or the content provided to the devices. It is understood that such databases may reside on any computer or computer system on network **N**, and that, in particular, the location is not limited to any particular server, for example.

[0035] Device **130** may be, for example and without limitation, a cellular telephone or a portable digital assistant, includes a network interface **131**, a memory **133**, a processor **135**, a screen **137**, and an input device **139**. Network interface **131** is used by device **130** to communication over a wireless network, such as a cellular telephone network, a WiFi network or a WiMax network, and then to other telephones through a public switched telephone network (PSTN) or to a satellite, or over the Internet. Memory **133** includes programming required to operate device **130** (such as an operating system or virtual machine instructions), and may include portions that store information or programming instructions obtained over network interface **131**, or that are input by the user (such as telephone numbers or images from a device camera (not shown). In one embodiment screen **137** is a touch screen, providing the functions of the screen and input device **139**.

[0036] Authoring platform **110** includes a computer or computer system having a memory **111**, a processor **113**, a screen **115**, and an input device **117**. It is to be understood that memory **111**, processor **113**, screen **115**, and input device **117** are configured such a program stored in the memory may be executed by the processor to accept input from the input device and display information on the screen. Further, the program stored in memory **111** may also instruct authoring platform **110** to provide programming or information, as indicated by the line labeled "A" and to receive information, as indicated by the line labeled "B."

[0037] Memory 111 is shown schematically as including a stored program referred to herein, and without limitation, as an authoring tool 112. In one embodiment, authoring tool 112 is a graphical system for designing the layout of features as a display that is to appear on screen 137. One example of authoring tool 112 is the CDER<sup>TM</sup> publishing platform (Express Mobile, Inc., Novato, CA).

#### WO 2009/126591

#### PCT/US2009/039695

[0038] In another embodiment, which is not meant to limit the scope of the present invention, device **130** may include an operating system having a platform that can interpret certain routines. Memory **111** may optionally include programming referred to herein, and without limitation, as routines **114** that are executable on device **130**.

[0039] Routines **114** may include device-specific routines - that is, codes that are specific to the operating system, programming language, or platform of specific devices **130**, and may include, but are not limited to, Java, Windows Mobile, Brew, Symbian OS, or Open Handset Alliance (OHA). Several examples and embodiments herein are described with reference to the use of Java. It is to be understood that the invention is not so limited, except as provided in the claims, and that one skilled in the art could provide Players for devices using routines provided on a platform. Thus as an example, routines **114** may include Java API's and an authoring tool System Development Kit (SDK) for specific devices **130**.

[0040] Server 120 is a computer or computer system that includes a network interface 121, a memory 123.and a processor 125. Is to be understood that network interface 121, memory 123, and processor 125 are configured such that a program stored in the memory may be executed by the processor to: accept input and/or provide output to authoring platform 110; accept input and/or provide output through network interface 121 over network N to network interface 131; or store information from authoring platform 110 or from device 130 for transmission to another device or system at a later time.

[0041] In one embodiment, authoring platform **110** permits a user to design desired displays for screen **137** and actions of device **130**. In other words, authoring platform **110** is used to program the operation of device **130**. In another embodiment, authoring platform **110** allows a user to provide input for the design of one or more device displays and may further allow the user to save the designs as device specific Applications. The Applications may be stored in memory **123** and may then be sent, when requested by device **130** or when the device is otherwise accessible, over network **N**, through network interface **130** for storage in memory **133**.

[0042] In an alternative embodiment, analytics information from devices **130** may be returned from device **130**, through network **N** and server **120**, back to authoring platform **110**, as indicated by line **B**, for later analysis. Analytics information includes, but is not limited to, user demographics, time of day, and location. The type of analytic content is only limited by which listeners have been activated for which objects and for which pages. Analytic content may include, but is not limited to, player-side page view, player-side forms-based content, player-side user interactions and player-side object status.
#### PCT/US2009/039695

[0043] Content server 140 is a computer or computer system that includes a network interface 141, a memory 143.and a processor 145. It is to be understood that network interface 141, memory 143, and processor 145 are configured such that a stored program in the memory may be executed by the processor to accepts requests **R** from device 130 and provide content **C** over a network, such as web server content the Internet, to device 130.

[0044] Figure 1B is schematic of an alternative embodiment system 100 for providing programming instructions to device 130 over a network N that is generally similar to the system of FIG. 1A. The embodiment of FIG. 1B illustrates that system 100 may include multiple servers 120 and/or multiple devices 130.

[0045] In the embodiment of FIG. 1B, system **100** is shown as including two or more servers **120**, shown illustratively and without limitation as servers **120a** and **120b**. Thus some of the programming or information between authoring platform **110** and one or more devices **130** may be stored, routed, updated, or controlled by more than one server **120**. In particular, the systems and methods described herein may be executed on one or more server **120**.

[0046] Also shown in FIG. 1B are a plurality of devices 130, shown illustratively and without limitation as device 130-1, 130-1, ... 130-N. System 100 may thus direct communication between individual server(s) 120 and specific device(s) 130.

[0047] As described subsequently, individual devices **130** may be provided with program instructions which may be stored in each device's memory **133** and where the instructions are executed by each device's processor **135**. Thus, for example, server(s) **120** may provide device(s) **130** with programming in response to the input of the uses of the individual devices. Further, different devices **130** may be operable using different sets of instructions, that is having one of a variety of different "device platforms." Differing device platforms may result, for example and without limitation, to different operating systems, different versions of an operating system, or different versions of virtual machines on the same operating system. In some embodiments, devices **130** are provided with some programming from authoring system **100** that is particular to the device.

[0048] In one embodiment, system 100 provides permits a user of authoring platform 110 to provide instructions to each of the plurality of devices 130 in the form of a device- or device-platform specific instructions for processor 135 of the device, referred to herein and without limitation as a "Player," and a device-independent program, referred to herein and without limitation as an "Application" Thus, for example, authoring platform 110 may be used to

### PCT/US2009/039695

generate programming for a plurality of devices **130** having one of several different device platforms. The programming is parsed into instructions used by different device platforms and instructions that are independent of device platform. Thus in one embodiment, device **130** utilizes a Player and an Application to execute programming from authoring platform **110**. A device having the correct Player is then able to interpret and be programmed according to the Application.

[0049] In one alternative embodiment, the Player is executed the first time by device **130** ("activated") through an Application directory. In another alternative embodiment, the Player is activated by a web browser or other software on device **130**. In yet another alternative embodiment, Player is activated through a signal to device **130** by a special telephone numbers, such as a short code.

[0050] When the Application and the Player are provided to memory **133**, the functioning of device **130** may occur in accordance with the desired programming. Thus in one embodiment, the Application and Player includes programming instructions which may be stored in memory **133** and which, when executed by processor **135**, generate the designed displays on screen **137**. The Application and Player may also include programming instructions which may be stored in memory **133** and which provide instructions to processor **135** to accept input from input device **139**.

[0051] Authoring tool **112** may, for example, produce and store within memory **111** a plurality of Players (for different devices **130**) and a plurality of Applications for displaying pages on all devices. The Players and Applications are then stored on one or more servers **120** and then provided to individual devices **130**. In general, Applications are provided to device **130** for each page of display or a some number of pages. A Player need be provided once or updated as necessary, and thus may be used to display a large number of Applications. This is advantageous for the authoring process, since all of the device-dependent programming is provided to a device only once (or possibly for some small number of upgrades), permitting a smaller Application, which is the same for each device **130**.

[0052] Thus, for example and without limitation, in one embodiment, the Player transforms device-independent instructions of the Application into device-specific instructions that are executable by device **130**. Thus, by way of example and without limitation, the Application may include Java programming for generating a display on screen **137**, and the Player may interpret the Java and instruct processor **135** to produce the display according to the Application for execution on a specific device **130** according to the device platform. The Application may in

### PCT/US2009/039695

general include, without limitation, instructions for generating a display on screen 137, instructions for accepting input from input device 139, instructions for interacting with a user of device 130, and/or instructions for otherwise operating the device, such as to place a telephone call.

[0053] The Application is preferably code in a device-independent format, referred to herein and without limitation as a Portable Description Language (PDL). The device's Player interprets or executes the Application to generate one or more "pages" ("Applications Pages") on a display as defined by the PDL. The Player may include code that is device-specific - that it, each device is provided with a Player that is used in the interpretation and execution of Applications. Authoring tool **112** may thus be used to design one or more device-independent Applications and may also include information on one or more different devices **130** that can be used to generate a Player that specific devices may use to generate displays from the Application.

[0054] In one embodiment, system 100 provides Players and Applications to one server 120, as in FIG. 1A. In another embodiment, system 100 provides Players to a first server 120a and Applications to a second server 120b, as in FIG. 1B.

[0055] In one embodiment, authoring tool **112** may be used to program a plurality of different devices **130**, and routines **114** may include device-specific routines. In another embodiment, the Player is of the type that is commonly referred to as a "thin client" – that is, software for running on the device as a client in client-server architecture with a device network which depends primarily on a central server for processing activities, and mainly focuses on conveying input and output between the user and the server.

[0056] In one embodiment, authoring platform **110** allows user to arrange objects for display on screen. A graphical user interface ("GUI," or "UI") is particularly well suited to arranging objects, but is not necessary. The objects may correspond to one or more of an input object, an output object, an action object, or may be a decorative display, such as a logo, or background color or pattern, such as a solid or gradient fill. In another embodiment, authoring platform **110** also permits a user to assign actions to one or more of an input object, an output object, or an action object. In yet another embodiment, authoring platform **110** also permits a user to bind one or more of an input object, an output object, or an action object with web services or web components, or permits a user to provide instructions to processor **135** to store or modify information in memory **133**, to navigate to another display or service, or to perform other actions, such as dialing a telephone number.

#### PCT/US2009/039695

[0057] In certain embodiments, the applicant model used in developing and providing Applications is a PDL. The PDL can be conceptually viewed as a device, operating system and virtual machine agnostic representation of Java serialized objects. In certain embodiments, the PDL is the common language for authoring tool **112**, the Application, and Player. Thus while either designing the Application with the authoring tool **112**, or programming with the SDK, the internal representation of the programming logic is in Java. In one embodiment the SDK is used within a multi-language software development platform comprising an IDE and a plug-in system to extend it, such as the Eclipse Integrated Development Environment (see, for example, http://www.eclipse.org/). At publish time the Java code is translated into a PDL. This translation may also occur in real-time during the execution of any Web Services or backend business logic that interacts with the user.

[0058] One embodiment for compacting data that may be used is described in co-pending U.S. Patent No. 6,546,397 to Rempell ("Rempell"), the contents of which are incorporated herein by reference. In that patent the compressed data is described as being a database. The terminology used here is a PDL, that is the "internal database" of Rempell is equivalent to the PDL of the present Application.

[0059] The use of a PDL, as described in Rempell, permits for efficient code and data compaction. Code, as well as vector, integer and Boolean data may be compacted and then compressed resulting in a size reduction of 40 to 80 times that of the original Java serialized objects. This is important not only for performance over the network but for utilizing the virtual memory manager of the Player more efficiently. As an example, the reassembled primitives of the Java objects may first undergo logical compression, followed by LZ encoding.

[0060] The use of a PDL also provides virtual machine and operating system independence. Since the reassembled primitives of the Application no longer have any dependencies from the original programming language (Java) that they were defined in. The PDL architecture takes full advantage of this by abstracting all the virtual machine and/or operating system interfaces from the code that processes the PDL.

[0061] In one embodiment, the PDL is defined by the means of nested arrays of primitives. Accordingly, the use of a PDL provides extensibility and compatibility, with a minimal amount of constraints in extending the Player seamlessly as market demands and device capabilities continue to grow. Compatibility with other languages is inherent based on the various Player abstraction implementations, which may be, for example and without limitation, Java CDC, J2SE or MIDP2 implementations.

#### PCT/US2009/039695

[0062] In one embodiment, the architecture of Player **P** includes an abstraction interface that separates all device, operating system and virtual machine dependencies from the Player's Application model business logic (that is, the logic of the server-side facilities) that extend the Application on the Player so that it is efficiently integrated into a comprehensive client/server Application. The use of an abstraction interface permits the more efficient porting to other operating systems and virtual machines and adding of extensions to the Application model so that a PDL can be implemented once and then seamlessly propagated across all platform implementations. The Application model includes all the currently supported UI objects and their attributes and well as all of the various events that are supported in the default Player. Further, less robust platforms can be augmented by extending higher end capabilities inside that platform's abstraction interface implementation.

[0063] In one embodiment, authoring platform **110** provides one or more pages, which may be provided in one Application, or a plurality of Applications, which are stored in memory **123** and subsequently provided to memory **133**. In certain embodiments, the Application includes instructions **R** to request content or web services **C** from content server **140**. Thus, for example and without limitation, the request is for information over the network via a web service, and the request **R** is responded to with the appropriate information for display on device **130**. Thus, for example, a user may request a news report. The Application may include the layout of the display, including a space for the news, which is downloaded form content server **140** for inclusion on the display. Other information that may be provided by content server **140** may include, but is not limited to, pages, Applications, multimedia, and audio.

[0064] Figure 2A is a schematic of a system 200 of an embodiment of system 100 illustrating the communications between different system components. System includes a response director 210, a web component registry 220, and a web service 230. System 200 further includes authoring platform 110, server 120, device 130 and content server 140 are which are generally similar to those of the embodiments of FIGS. 1A and 1B, except as explicitly noted.

[0065] Response director **210** is a computer or computer system that may be generally similar to server **120** including the ability to communicate with authoring platform **110** and one or more devices **130**. In particular, authoring platform **110** generates one or more Players (each usable by certain devices **130**) which are provided to response director **210**. Devices **130** may be operated to provide response director **210** with a request for a Player and to receive and install the Player. In one embodiment, device **130** provides response director **210** with device-specific information including but not limited to make, model, and/or software version of the device. Response

### PCT/US2009/039695

director **210** then determines the appropriate Player for the device, and provides the device with the Player over the network.

[0066] Web service **230** is a plurality of services obtainable over the Internet. Each web service is identified and/or defined as an entry in web component registry **230**, which is a database, XML file, or PDL that exists on a computer that may be a server previously described or another server **120**. Web component registry **230** is provided through server **120** to authoring platform **110** so that a user of the authoring platform may bind web services **230** to elements to be displayed on device **130**, as described subsequently.

[0067] In one embodiment, authoring platform **110** is used in conjunction with a display that provides a WYSIWYG environment in which a user of the authoring platform can produce an Application and Player that produces the same display and the desired programming on device **130**. Thus, for example, authoring tool **112** provides a display on screen **115** that corresponds to the finished page that will be displayed on screen **137** when an Application is intercepted, via a Player, on processor **135** of device **130**.

[0068] Authoring platform **110** further permits a user of the authoring platform to associate objects, such as objects for presenting on screen **137**, with components of one or more web services **230** that are registered in web component registry **220**. In one embodiment, information is provided in an XML file to web component registry **220** for each registered components of each web service **230**. Web component registry **220** may contain consumer inputs related to each web service **230**, environmental data such as PIM, time or location values, persistent variable data, outputs related to the web service, and/or optional hinting for improving the user's productivity.

[0069] A user of authoring platform **110** of system **200** may define associations with web services as WebComponent Bindings. In one embodiment, authoring platform **110** allows a user to associate certain objects for display that provide input or output to components of web service **230**. The associated bindings are saved as a PDL in server **120**.

[0070] In one embodiment, an XML web component registry **220** for each registered web service **230** is loaded into authoring platform **110**. The user of system **200** can then assign components of any web service **230** to an Application without any need to write code. In one embodiment, a component of web service **230** is selected from authoring platform **110** which presents the user with WYSIWYG dialog boxes that enable the binding of all the inputs and outputs of component of web service **230** to a GUI component of the Application as will be

### PCT/US2009/039695

displayed on screen 137. In addition, multiple components of one or more web service 230 can be assigned to any Object or Event in order to facilitate mashups. These Object and/or Event bindings, for each instance of a component of any web service 230, are stored in the PDL. The content server 140 handles all communication between device 130 and the web service 230 and can be automatically deployed as a web application archive to any content server.

[0071] Device 130, upon detecting an event in which a component of a web service 230 has been defined, assembles and sends all related inputs to content server 240, which proxies the request to web service 230 and returns the requested information to device 130. The Player on device 130 then takes the outputs of web service 230 and binds the data to the UI components in the Application, as displayed on screen 137.

[0072] In one embodiment, the mechanism for binding the outputs of the web service to the UI components is through symbolic references that matches each output to the symbolic name of the UI component. The outputs, in one embodiment, may include meta-data which could become part of the inputs for subsequent interactions with the web service.

[0073] For example, if a user of authoring platform **110** wants to present an ATOM feed on device **130**, they would search through a list of UI Components available in the authoring platform, select the feed they want to use, and bind the output of the feed summary to a textbox. The bindings would be saved into the PDL on server **120** and processed by device **130** at runtime. If the ATOM feed does not exist a new one can be added to the web component registry that contains all the configuration data required, such as the actual feed URL, the web component manager URL, and what output fields are available for binding.

[0074] In another embodiment, components of web services **230** are available either to the user of authoring platform **110** or otherwise accessible through the SDK and Java APIs of routines **114**. System **200** permits an expanding set of components of web services **230** including, but not limited to: server pages from content server **120**; third-party web services including, but not limited to: searching (such through Google or Yahoo), maps (such as through MapQuest and Yahoo), storefronts (such as through ThumbPlay), SMS share (such as through clickatel), stock quotes, social networking (such as through FaceBook), stock quotes, weather (such as through Accuweather) and/or movie trailers. Other components include web services for communication and sharing through chats and forums and rich messaging alerts, where message alerts are set-up that in turn could have components of Web Services **230** defined within them, including the capture of consumer generated and Web Service supplied rich media and textual content.

### PCT/US2009/039695

[0075] System **200** also permits dynamic binding of real-time content, where the inputs and outputs of XML web services are bound to GUI components provided on screen **137**. Thus, for example, a user of authoring platform **110** may bind attributes of UI Objects to a particular data base field on a Server. When running the Application, the current value in the referenced data base will be immediately applied. During the Application session, any other real time changes to these values in the referenced data base will again be immediately displayed.

[0076] As an example of dynamic binding of real-time content, an RSS feeds and other forms of dynamic content may be inserted into mobile Applications, such as device **130**, using system **200**. Authoring platform **110** may include a "RSS display" list which permits a user to select RSS channels and feeds from an extensible list of available dynamic content. Meta data, such as titles, abstracts and Images can be revealed immediately by the user as they traverse this RSS display list, bringing the PC experience completely and conveniently to mobile devices **130**. In addition, Authoring platform **110** may include a dialog box that dynamically links objects to data and feeds determined by RSS and chat databases. Any relevant attribute for a page view and/or object can be dynamically bound to a value in a server-side database. This includes elements within complex objects such as: any icon or text element within a graphical list; any icon within a launch strip; any feature within any geographical view of a GIS service object; and/or any virtual room within a virtual tour.

[0077] As an example of third-party web services 230 that may be provided using system 200, a user of authoring platform 110 can place, for example, Yahoo maps into device 130 by binding the required component of the Yahoo Maps Web Service, such as Yahoo Map's Inputs and/or Outputs to appropriate Objects of authoring platform 110. System 200 also provides binding to web services for text, image and video searching by binding to components of those web services.

[0078] In one embodiment, an Application for displaying on device **130** includes one or more Applications Pages, each referred to herein as an "XSP," that provides functionality that extends beyond traditional web browsers. The XSP is defined as a PDL, in a similar manner as any Application, although it defines a single page view, and is downloaded to the Player dynamically as required by the PDL definition of the Application. Thus, for example, while JSPs and ASPs, are restricted to the functionality supported by the web browser, the functionality of XSPs can be extended through authoring platform **110** having access to platform dependent routines **114**, such as Java APIs. Combined with dynamic binding functionality, an XSP, a page can be saved as a page object in an author's "pages" library, and then can be dynamically populated with real-time

### PCT/US2009/039695

content simultaneously as the page is downloaded to a given handset Player based on a newly expanded API. XSP Server Pages can also be produced programmatically, but in most cases authoring platform **110** will be a much more efficient way to generate and maintain libraries of dynamically changing XSPs.

[0079] With XSPs, Applications Pages that have dynamic content associated with them can be sent directly to device **130**, much like how a web browser downloads an HTML page through a external reference. Without XSPs, content authors would have to define each page in the Application. With XSPs, no pages need to be defined. Thus, for example, in a World Cup Application, one page could represent real-time scores that change continuously on demand. With polling (for example, a prompt to the users asking who they predict will win a game), a back-end database would tabulate the information and then send the results dynamically to the handsets. With a bar chart, the Application would use dynamic PDL with scaling on the fly. For example, the server would recalibrate the bar chart for every ten numbers.

[0080] Other combinations of components of web services 230 include, but are not limited to, simultaneous video chat sessions, inside an integrated page view, with a video or television station; multiple simultaneous chat sessions, each with a designated individual and/or group, with each of the chat threads visible inside an integrated page view.

[0081] Another extension of an XSP is a widget object. Widgets can be developed from numerous sources including, but not limited to, authoring platform 110, a Consumer Publishing Tool, and an XML to Widget Conversion Tool where the SDK Widget Libraries are automatically populated and managed, or Widget Selection Lists that are available and can be populated with author defined Icons.

## APPLICATIONS, PLAYERS, AND PROCESSING IN A DEVICE

[0082] Figure 2B is a schematic of one embodiment of a device **130** illustrating an embodiment of the programming generated by authoring platform **110**. Memory **133** may include several different logical portions, such as a heap **133a**, a record store **133b** and a filesystem (not shown).

[0083] As shown in FIG. 2B, heap **133a** and record store **133b** include programming and/or content. In general, heap **133a** is readily accessible by processor **135** and includes, but is not limited to portions that include the following programming: a portion **133a1** for virtual machine compliant objects representing a single Page View for screen **137**; a portion **133a2** for a Player; a portion **133a3** for a virtual machine; and a portion **133a4** for an operating system.

### PCT/US2009/039695

[0084] Record store 133b (or alternatively the filesystem) includes, but is not limited to, portions 133b1 for Applications and non-streaming content, which may include portions 133a2 for images, portions 133a4 for audio, and/or portions 133a5 for video. and portions 133b2 for non-Application PDLs, such as a Master Page PDL for presenting repeating objects, and Alerts, which are overlayed on the current page view. Other content, such as streaming content may be provided from network interface 131 directly to the Media Codec of device 130 with instructions from Player on how to present the audio or video.

[0085] In one embodiment, the Player includes a Threading Model and a Virtual Memory Manager. The Threading Model first manages a queue of actions that can be populated based on Input/Output events, Server-side events, time-based events, or events initiated by user interactions. The Threading Model further manages the simultaneous execution of actions occurring at the same time. The Virtual Memory Manager includes a Logical Virtual Page controller that provides instructions from the record store to the heap, one page at time. Specifically, the Virtual Memory Manager controls the transfer of one of the Application Pages and its virtual machine compliant objects into portion **133a1** as instructions readable by the Player or Virtual Machine. When the Player determines that a new set of instructions is required, the information (such as one Application Page is retrieve from the Record store, converted into virtual machine compliant objects (by processor **135** and according to operation by the Player, Virtual Machine, etc). and stored in heap **133a**. Alternatively, the Player may augment virtual machine compliant objects with its own libraries for managing user interactions, events, memory, etc.

[0086] The connection of portions 133a1, 133a2, 133a3, 133a4, record store 133b and processor 135 are illustrative of the logical connection between the different types of programming stored in Heap 133a and record store 133b, that is, how data is processed by processor 135.

[0087] The Player determines which of the plurality of Application Pages in portion **133b1** is required next. This may be determined by input actions from the Input Device **139**, or from instructions from the current Application Page. The Player instructs processor **135** to extract the PDF from that Applications Page and store it in portion **133a1**. The Player then interprets the Application Page extracted from PDL which in turn defines all of the virtual machine compliant Objects, some of which could have attributes that refer to images, audio, and/or video stored in portions **133a3**, **133a4**, **133a5**, respectively.

[0088] The Virtual Machine in portion **133a3** processes the Player output, the Operating System in portion **133a3** processes the Virtual Machine output which results in machine code that is processed by the Operating System in portion **133a4**.

[0089] In another embodiment, the Player is a native program that interacts directly with the operating system.

# EMBODIMENTS OF A PUBLISHING ENVIRONMENT

[0090] In one embodiment, authoring platform **110** includes a full-featured authoring tool **112** that provides a what-you-see-is-what-you-get (WYSIWYG) full featured editor. Thus, for example, authoring tool **112** permits a user to design an Application by placing objects on canvas **305** and optionally assigning actions to the objects and save the Application. System **100** then provides the Application and Player to a device **130**. The Application as it runs on device **130** has the same look and operation as designed on authoring platform **110**. In certain embodiments, authoring platform **110** is, for example and without limitation, a PC-compatible or a Macintosh computer.

[0091] Authoring platform **110** produces an Application having one or more Applications Pages, which are similar to web pages. That is, each Applications Page, when executed on device **130** may, according to its contents, modify what is displayed on screen **137** or cause programming on the device to change in a manner similar to how web pages are displayed and navigated through on a website.

[0092] In one embodiment, authoring tool **112** allows a user to place one or more objects on canvas **305** and associate the objects with an Applications Pages. Authoring platform **110** maintains a database of object data in memory **111**, including but not limited to type of object, location on which page, and object attributes. The user may add settings, events, animations or binding to the object, from authoring tool **112**, which are also maintained in memory **111**. Authoring tool **112** also allows a user to define more than one Applications Page.

[0093] In another embodiment, authoring tool **112**, provides Java programming functions of the Java API for specific devices **130** as pull-down menus, dialog boxes, or buttons. This permits a user of authoring platform **110** to position objects that, after being provided as an Application to device **130**, activate such Java functions on the device.

[0094] In certain embodiments, authoring platform **110**, as part of system **100**, permits designers to include features of advanced web and web services Applications for access by users

### PCT/US2009/039695

of device **130**. Some of the features of advanced web and web services include, but are not limited to: slide shows, images, video, audio, animated transitions, multiple chats, and mouse interaction; full 2-D vector graphics; GIS (advanced LBS), including multiple raster and vector layers, feature sensitive interactions, location awareness, streaming and embedded audio/video, virtual tours, image processing and enhancement, and widgets. In other embodiments the features are provided for selection in authoring platform **110** through interactive object libraries.

[0095] In certain embodiments, authoring platform **110**, as part of system **100**, allows the inclusion of child objects which may eventually be activated on device **130** by the user of the device or by time. The uses of the child objects on device **130** include, but are not limited to: mouse over (object selection), hover and fire events and launching of object-specific, rich-media experiences.

[0096] In certain other embodiments, authoring platform **110**, as part of system **100**, provides advanced interactive event models on device **130**, including but not limited to: user-, time- and/or location-initiated events, which allow content developers to base interactivity on specific user interactions and/or instances in time and space; timelines, which are critical for timing of multiple events and for animations when entering, on, or exiting pages of the Application; waypoints, which act similar to key frames, to allow smooth movement of objects within pages of the Application. Waypoints define positions on a page object's animation trajectory. When an object reaches a specific waypoint other object timelines can be initiated, creating location-sensitive multiple object interaction, and/or audio can be defined to play until the object reaches the next waypoint.

[0097] Authoring platform **110** may also define a Master Page, which acts as a template for an Applications Page, and may also define Alert Pages, which provide user alerts to a user of device **130**.

[0098] In certain embodiments, authoring platform **110**, as part of system **100**, provides full style inheritance on device **130**. Thus, for example and without limitation, both master page inheritance (for structural layout inheritance and repeating objects) and object styles (for both look and feel attribute inheritance) are supported. After a style has been defined for an object, the object will inherit the style. Style attributes include both the look and the feel of an object, including mouse interaction, animations, and timelines. Each page may include objects that may be a parent object or a child object. A child object is one that was created by first selecting a parent object, and then creating a child object. Child objects are always part of the same drawing layer as its parent object, but are drawn first, and are not directly selectable when running the

## PCT/US2009/039695

Application. A parent object is any object that is not a child object, and can be selected when running the Application.

[0099] As an example, the user of authoring tool **112** may create various text objects on canvas **305** using a style that sets the font to red, the fonts of these objects will be red. Suppose user of authoring tool **112**changes the font color of a specific button to green. If later, the user of authoring tool **112**changes the style to blue; all other text objects that were created with that style will become blue except for the button that had been specifically set to green.

[0100] In certain other embodiments, authoring platform **110** provides page view, style, object, widget and Application template libraries. Authoring platform **110** may provide templates in private libraries (available to certain users of the authoring platform) and public libraries (available to all users of the authoring platform). Templates may be used to within authoring platform **110** to define the look and feel of the entire Application, specific pages, or specific slide shows and virtual tours a seen on device 130.

[0101] Figures 3A and 3B illustrate one embodiment of a publisher interface **300** as it appears, for example and without limitation, on screen **115** while executing authoring tool **112**. In one embodiment, publisher interface **300** includes a Menu bar **301**, a Tool bar **303**, a Canvas **305**, a Layer Inspector **307** having subcomponents of a page/object panel **307a**, an object style panel **307b**, and a page alert panel **307c**, and a Resource Inspector **309**.

[0102] In general, publisher interface **300** permits a user of authoring platform **110** to place objects on canvas **305** and then associate properties and/or actions to the object, which are stored in the Application. As described subsequently, publisher interface **300** permits a user to program a graphical interface for the screen **137** of device **130** on screen **115** of authoring platform **110**, save an Application having the programming instructions, and save a Player for the device. The intended programming is carried out on device **130** when the device, having the appropriate device platform Player, receives and executes the device-independent Application.

[0103] Thus, for example, authoring tool **112** maintains, in memory **111**, a list of every type of object and any properties, actions, events, or bindings that may be assigned to that object. As objects are selected for an Application, authoring tool **112** further maintains, in memory **111**, a listing of the objects. As the user selects objects, publisher interface **300** provides the user with a choice of further defining properties, actions, events, or bindings that may be assigned to each particular object, and continues to store the information in memory **111**.

#### PCT/US2009/039695

[0104] In one embodiment, publisher interface **300** is a graphical interface that permits the placement and association of objects in a manner typical of, for example, vector graphics editing programs (such as Adobe Illustrator). Objects located on canvas **305** placed and manipulated by the various commands within publisher interface **300** or inputs such as an input device **117** which may be a keyboard or mouse. As described herein, the contents of canvas **305** may be saved as an Application that, through system **100**, provide the same or a similar placement of objects on screen **137** and have actions defined within publisher interface **300**. Objects placed on canvas **305** are intended for interaction with user of device **130** and are referred to herein, without limitation, as objects or UI (user interface) objects. In addition, the user of interface **300** may assign or associate actions or web bindings to UI objects placed on canvas **305** with result in the programming device **130** that cause it to respond accordingly.

[0105] Objects include, but are not limited to input UI objects, response UI objects. Input UI objects include but are not limited to: text fields (including but not limited to alpha, numeric, phone number, or SMS number); text areas; choice objects (including but not limited to returning the selected visible string or returning a numeric hidden attribute); single item selection lists (including but not limited to returning the selected visible string or returning a numeric hidden attribute); multi item selection lists (including but not limited to returning all selected items (visible text string or hidden attribute) or cluster item selection lists (returning the hidden attributes for all items).

[0106] Other input UI objects include but are not limited to: check boxes; slide show (including but not limited to returning a numeric hidden attribute, returning a string hidden attribute, or returning the hidden attributes for all slides); and submit function (which can be assigned to any object including submit buttons, vectors, etc.).

[0107] Response UI Objects may include, but are not limited to: single line text objects, which include: a text Field (including but not limited to a URL, audio URL, or purchase URL), a text button, a submit button, or a clear button. Another response UI objects include: a multiple line text object, which may include a text area or a paragraph; a check box; an image; a video; a slide show (with either video or image slides, or both); choice objects; list objects; or control lists, which control all the subordinate output UI objects for that web component. Control list objects include, but are not limited to: list type or a choice type, each of which may include a search response list or RSS display list.

# PCT/US2009/039695

[0108] As a further example of objects that may be used with authoring tool **112**, Table I lists Data Types, Preferred Input, Input Candidates, Preferred Output and Output Candidates for one embodiment of an authoring tool.

Data Types	Preferred Input	Input Candidates	Preferred Output	Output Candidates
boolean	Check Box	Check Box	Check Box	Check Box
Int	Text Field (integer)	Text Field (integer) Text Field (Phone #) Text Field (SMS #) Choice List (single select)	Text Field (integer)	Text Field (integer) Text Field (Phone #) Text Field (SMS #) Choice List (single select) Text Button
String	Text Field (Alpha)	Any	Text Field (Alpha)	Any
multilineString	Text Area	Text Area	Text Area	Text Area Paragraph
ImageURL	N/A	N/A	Image	Image Slide Show
VideoURL	N/A	N/A	Video	Video Slide Show
List	Single Item List	Single Item List Multi-Select List Complex List Choice Slide Show	Single Item List	Any List Type Any Choice Type (see Complex List Specification)
ComplexList	Complex List	Single Item List Multi-Select List Complex List	Single Item List	Any List Type (see Complex List Specification)
Slideshow	Slide Show	Slide Show	Slide Show	Slide Show
SearchResponseList	N/A	N/A	Search Response List	Search Response List Control List Complex List Choice
RSSList	N/A	N/A	RSS Display List	RSS Display List Control List Complex List Choice
SingleSelectionList	Choice	Choice Complex List	Choice	Choice Complex List
MultiSelectionList	Multi-Selection List	Multi-Selection List	Multi-Selection List	Multi-Selection List
ServiceActivation	Submit Button	Any	N/A	N/A
ChannellmageURL	N/A	N/A	lmage	Image Video Slide Show
ChannelDescription	N/A	N/A	Text Area	Text Area Paragraph Text Field Text Button List Choice

ChannelTitle	N/A	N/A	Text Field	Text Field
onumornae	1477			Text Button
				Paragraph
				Text Area
				List
				Choice
IR			Text Field	Text Field
One		4 4 7 7	(IIR) request)	(LIRI request)
Audio HRI			Tovt Field	Tavt Field
Addio One			(Audio UPL request)	(Audio LIDI request)
Purchasa LIDI			Toxt Eigld	Toxt Field
FURCHASE UNL			(Developed UD)	IEXL FIELD
			(Purchase UKL request)	(Purchase UKL request)
Image Data			Image	Image
				Slide Show
Image List Data			Slide Show	Slide Show
				Image
Persistent Variable	N/A	N/A	N/A	N/A
Pipeline Multiple Select	Multi-select List	Multi-select List	N/A	N/A
		Complex List		
		Slide Show		
Phone Number	Text Field	Text Field	Text Field	Text Field
	(numeric type)	Text Button	(numeric type)	Text Button
Hidden Attribute	Complex List	Complex List	Complex List	Complex List
, wash / kalboto	o o mpion and c	Slide Show	o on prost aloc	Slide Show
Collection List	NI/A	N/A	Slide Show	Complex List
CONCOUNT LISE	11/ N	IVA		Clide Chow
				Silve Show

# Table I. One embodiment of supported objects

[0109] In general, publisher interface **300** permits a user to define an Application as one or more Applications Pages, select UI objects from Menu bar **301** or Tool bar **303** and arrange them on an Applications Page by placing the objects canvas **305**. An Application Page is a page that is available to be visited through any navigation event. Application Pages inherit all the attributes of the Master Page, unless that attribute is specifically changed during an editing session.

[0110] Authoring platform **110** also stores information for each UI object on each Application Page of an Application. Layer Inspector **307** provides lists of Applications Pages, UI objects on each Applications Page, and Styles, including templates. Objects may be selected from canvas **305** or Layer Inspector **307** causing Resource Inspector **309** to provide lists of various UI objects attributes which may be selected from within the Resource Inspector. Publisher interface **300** also permits a user to save their work as an Application for layer transfer and operation of device **130**. Publisher interface **300** thus provides an integrated platform for designing the look and operation of device **130**.

[0111] The information stored for each UI object depends, in part, on actions which occur as the result of a user of device **130** selecting the UI object from the device. UI objects include, but

#### PCT/US2009/039695

are not limited to: navigational objects, such as widget or channel launch strips or selection lists; message objects for communicating, such as a multiple chat, video chat, phone and/or SMS lists or fields or a pop-up alert; text fields or areas; check boxes; pull down menus; selection lists and buttons; pictures; slide shows; video or LBS maps; shapes or text defined by a variety of tools; a search response; or an RSS display.

[0112] In certain embodiments, publisher interface **300** permits a user to assign action to UI objects, including but not limited to, programming of the device **130** or a request for information over network **N**. In one embodiment, for example and without limitation, publisher interface **300** has a selection to bind a UI object to a web service - that is, associate the UI object or a manipulation or selection of UI object with web services. Publisher interface **300** may also include many drawing and text input functions for generating displays that may be, in some ways, similar to drawing and/or word processing programs, as well as toolbars and for zooming and scrolling of a workspace.

[0113] Each UI object has some form, color, and display location associate with it. Further, for example and without limitation, UI objects may have navigational actions (such as return to home page), communications actions (such as to call the number in a phone number field), or web services (such as to provide and/or retrieve certain information from a web service). Each of the these actions requires authoring platform **110** to store the appropriate information for each action. In addition, UI objects may have associated patent or child objects, default settings, attributes (such as being a password or a phone number), whether a field is editable, animation of the object, all of which may be stored by authoring platform **110**, as appropriate.

[0114] Menu bar **301** provides access features of publisher interface **300** through a series of pull-down menus that may include, but are not limited to, the following pull-down menus: a File menu **301a**, an Edit menu **301b**, a View menu **301c**, a Project menu **301d**, an Objects menu **301e**, an Events menu **301f**, a Pages menu **301g**, a Styles menu **301h**, and a Help menu **301i**.

[0115] File menu **301a** provides access to files on authoring platform **110** and may include, for example and without limitation, selections to open a new Application or master page, open a saved Application, Application template, or style template, import a page, alert, or widget, open library objects including but not limited to an image, video, slide show, vector or list, and copying an Application to a user or to Server **120**.

[0116] Edit menu **301b** may include, but is not limited to, selections for select, cut, copy, paste, and edit functions.

### PCT/US2009/039695

[0117] View menu **301c** may include, but is not limited to, selections for zooming in and out, previewing, canvas **305** grid display, and various palette display selections.

[0118] Project menu **301d** may include, but is not limited to, selections related to the Application and Player, such as selections that require a log in, generate a universal Player, generate server pages, activate server APIs and extend Player APIs. A Universal Player will include all the code libraries for the Player, including those that are not referenced by the current Application. Server APIs and Player APIs logically extend the Player with Server-side or deviceside Application specific logic.

[0119] Objects menu **301e** includes selections for placing various objects on canvas **305** including, but not limited to: navigation UI objects, including but not limited to widget or channel launch strips or selection lists; message-related UI objects, including but not limited to multiple chat, video chat, phone and/or SMS lists or fields, or a pop-up alert; shapes, which provides for drawing tools; forms-related objects, including but not limited to text fields; scrolling text box, check box, drop-down menu, list menu, submit button or clear button; media-related UI objects such as pictures, slide shows, video or LBS maps; text-related UI objects such as buttons or paragraphs; and variables, including but not limited to time, date and audio mute control.

[0120] Events menu **301f** includes selections for defining child objects, mouse events, animations or timelines.

[0121] Pages menu **301g** includes selection for handling multi-page Applications, and may include selections to set a master page, delete, copy, add or go to Applications Pages.

[0122] Styles menu **301h** includes selections to handle styles, which are the underlying set of default appearance attributes or behaviors that define any object that is attached to a style. Styles are a convenient way for quickly creating complex objects, and for changing a whole collection of objects by just modifying their common style. Selections of Styles menu **301h** include, but not limited to, define, import, or modify a style, or apply a template. Help menu **301i** includes access a variety of help topics.

[0123] Tool bar **303** provides more direct access to some of the features of publisher interface **300** through a series of pull-down menus. Selections under tool bar **303** may include selections to:

• control the look of publisher interface 300, such as a Panel selection to control the

for hiding or viewing various panels on publisher interface 300;

[0124] control the layout being designed, such as an Insert Page selection to permit a user to insert and name pages;

[0125] control the functionality of publisher interface **300**, such as a Palettes selection to choose from a variety of specialized palettes, such as a View Palette for zooming and controlling the display of canvas **305**, a Command Palette of common commands, and Color and Shape Palettes;

[0126] place objects on canvas **305**, which may include selections such as: a Navigation selection to place navigational objects, such as widget or channel launch strips or selection lists), a Messages selection to place objects for communicating, such as a multiple chat, video chat, phone and/or SMS lists or fields, or a pop-up alert, a Forms selection to place objects such as text fields or areas, check boxes, pull down menus, selection lists, and buttons, a Media selection to place pictures, slide shows, video or LBS maps, and a Shapes selection having a variety of drawing tools, a Text selection for placing text, a search response, or an RSS display, and Palettes.

[0127] In one embodiment, Tool bar **303** includes a series of pull-down menus that may include, but are not limited to, items from Menu bar **301** organized in the following pull-down menus: a Panel menu **303a**, an Insert Page menu **303b**, a Navigation menu **303c**, a Messages menu **303d**, a Forms menu **303e**, a Media menu **303f**, a Shapes menu **303g**, a Text menu **303h**, and a Palettes menu **301i**.

[0128] Panel menu **303a** permits a user of authoring platform **110** to change the appearance of interface **300** by, controlling which tools are on the interface or the size of canvas **305**. Insert Page menu **303b** permits a user of authoring platform **110** to open a new Application Page. Navigation menu **303c** displays a drop down menu of navigational-related objects such as a widget or channel launch strip or selection list. Messages menu **303d** displays a drop down menu of messaging-related objects such as multiple chat, video chat, phone or SMS lists or fields, and pop-up alerts. Forms menu **303e** displays a drop down menu of forms-related objects including, but not limited to, a text field, a text area, a check box, a drop down menu, a selection list, a submit button, and a clear button. Media menu **303f** displays a drop down menu of media-related objects including, but not limited to, a picture, slide show, video or LBS map. Shapes menu **303g** displays a drop down menu of draw tools, basic shapes, different types of lines and arrows and access to a shape library. Text menu **303j** displays a drop down menu of text-related objects,

## PCT/US2009/039695

including but not limited to a text button, paragraph, search response, RSS display and variables such as time and date.

[0129] Palettes menu **301i** includes a selection of different palettes that can be moved about publisher interface **300**, where each palette has specialized commands for making adjustments or associations to objects easier. Palettes include, but are not limited to: a page view palette, to permit easy movement between Applications Pages; a view palette, to execute an Application or zoom or otherwise control the viewing of an Application; a commands palette having editing commands; a color palette for selection of object colors; and a shapes palette to facilitate drawing objects.

[0130] Layer inspector **307** permits a user of publisher interface **300** to navigate, select and manipulate UI objects on Applications Pages. Thus, for example, a Page/objects panel **307a** of layer inspector **307** has a listing that may be selected to choose an Applications Pages within and Application, and UI objects and styles within an Applications Page. An Object styles panel **307b** of layer inspector **307** displays all styles on the Applications Page and permits selection of UI objects for operations to be performed on the objects.

[0131] Thus, for example, when objects from Menu bar **301** or Tool bar **303** are placed on canvas **305**, the name of the object appears in Page/objects panel **307a**. Page/objects panel **307a** includes a page display **307a1** and an objects display **307a2**. Page display **307a1** includes a pull down menu listing all Applications Pages of the Application, and objects display **307a2** includes a list of all objects in the Applications Page (that is, objects on canvas **305**).

[0132] In general, page/objects panel **307a** displays various associations with a UI object and permits various manipulations including, but not limited to, operations for parent and child objects that are assigned to a page, and operations for object styles, and permits navigating between page types and object styles, such as switching between the master page and Application pages and deselecting object styles and alerts, opening an Edit Styles Dialog Box and deselecting any master, Application or alert page, or selecting an alert page and deselecting any Master Page or Application Page. A parent or child object can also be selected directly from the Canvas. In either case, the Resource Inspector can then be used for modifying any attribute of the selected object.

[0133] Examples of operations provided by page/objects panel **307a** on pages include, but are not limited to: importing from either a user's private page library or a public page library; deleting a page; inserting a new page, inheriting all the attributes of the Master Page, and placing

### PCT/US2009/039695

the new page at any location in the Page List; editing the currently selected page, by working with an Edit Page Dialog Box. While editing all the functions of the Resource Inspector **309** are available, as described subsequently, but are not applied to the actual page until completing the editing process.

[0134] Examples of operations provided by of page/objects panel **307a** on objects, which may be user interface (UI) objects, include but are not limited to: changing the drawing order layer to: bring to the front, send to the back, bring to the front one layer, or send to the back one layer; hiding (and then reshowing) selected objects to show UI objects obstructed by other UI Objects, delete a selected UI Page Object, and editing the currently selected page, by working with a Edit Page Dialog Box.

[0135] Object styles panel **307b** of layer inspector **307** displays all styles on the Applications Page and permits operations to be performed on objects, and is similar to panel **307a**. Examples of operations provided by object style panel **307b** include, but are not limited to: importing from either a user's private object library or a public object library; inserting a new object style, which can be inherited from a currently selected object, or from a previously defined style object; and editing a currently selected object style by working with an Edit Style Dialog Box.

[0136] Style attributes can be assigned many attributes, including the look, and behavior of any object that inherits these objects. In addition, List Layout Styles can be created or changed as required. A layout style can define a unbounded set of Complex List Layouts, including but not limited to: the number of lines per item in a list, the number of text and image elements and their location for each line for each item in the last, the color and font for each text element, and the vertical and horizontal offset for each image and text element.

[0137] Alerts Panel **307c** provides a way of providing alert pages, which can have many of the attributes of Application Pages, but they are only activated through an Event such as a user interaction, a network event, a timer event, or a system variable setting, and will be superimposed onto whatever is currently being displayed. Alert Pages all have transparent backgrounds, and they function as a template overlay, and can also have dynamic binding to real time content.

[0138] Resource inspector **309** is the primary panel for interactively working with UI objects that have been placed on the Canvas **305**. When a UI object is selected on Canvas 305, a user of authoring platform **110** may associate properties of the selected object by entering or selecting from resource inspector **309**. In one embodiment, resource inspector **309** includes five tab

## PCT/US2009/039695

selections: Setting Tab **309a**, Events Tab **309b**, Animation Tab **309c**, Color Tab **309d** which includes a color palette for selecting object colors, and Bindings Tab **309e**.

[0139] Settings Tab **309a** provides a dialog box for the basic configuration of the selected object including, but not limited to, name, size, location, navigation and visual settings. Depending upon the type of object, numerous other attributes could be settable. As an example, the Setting Tab for a Text Field may include dialog boxes to define the text field string, define the object style, set the font name, size and effects, set an object name, frame style, frame width, text attributes (text field, password field, numeric field, phone number, SMS number, URL request).

[0140] As an example of Setting Tab **309a**, FIG. 3B shows various selections including, but not limited to, setting **309a1** for the web page name, setting **309a2** for the page size, including selections for specific devices **130**, setting **309a3** indicating the width and height of the object, and setting **309a4** to select whether background audio is present and to select an audio file.

[0141] Figure 3C illustrates an embodiment of the Events Tab **309b**, which includes all end user interactions and time based operations. The embodiment of Events Tab **309b** in FIG. 3C includes, for example and without limitation, an Events and Services **309b1**, Advanced Interactive Settings **309b2**, Mouse State **309b3**, Object Selected Audio Setting **309b4**, and Work with Child Objects and Mouse Overs button **309b5**.

[0142] Events and Services **309b1** lists events and services that may be applied to the selected objects. These include, but are not limited to, going to external web pages or other Applications pages, either as a new page or by launching a new window, executing an Application or JavaScript method, pausing or exiting, placing a phone call or SMS message, with or without single or multiple Player download, show launch strip, or go back to previous page. Examples of events and services include, but are not limited to those listed in Table II

Goto External Web Page replacing Current Frame	ChoiceObject : Remove Icon from Launch	
	Strip	
Goto External Web Page Launched in a New	Goto a specific Internal Web Page with	
Window	Alert. "Backend Synchronization"	
Goto a specific Internal Web Page	Goto Widget Object	
Goto the next Internal Web Page	Generate Alert. "With a Fire Event"	
Goto External Web Page replacing the Top Frame	Send SMS Message from Linked Text Field	
Execute JavaScript Method	Toggle Alert. "Display OnFocus, Hide OffFocus"	
Pause/Resume Page Timeout	Execute an Application with Alert. "With a Fire Event"	
Execute an Application	Goto Logical First Page	
Goto a specific Internal Web Page with setting	Generate Alert with Backend	
starting slide	Synchronization	
Exit Application	Send SMS Message with Share (Player	
	Download)	
Exit Player	Place PhoneCall from linked Text Field	
	with Share (Player Download)	
Place PhoneCall from linked Text Field	Send IM Alert from linked Text Field or	
	Text Area	
Text Field/Area: Send String on FIRE	Set and Goto Starting Page	
ChoiceObject : Add Icon to Launch Strip	Populate Image	
Text Field/Area: Send String on FIRE or Numeric	Preferred Launch Strip	
Keys		

# Table II. Events and Services

[0143] Advanced Interactive Settings **309b2** include Scroll Activation Enabled, Timeline Entry Suppressed, Enable Server Listener, Submit Form, Toggle Children on FIRE, and Hide Non-related Children, Mouse State **309b3** selections are Selected or Fire. When Mouse State Selected is chosen, Object Selected Audio Setting **309b4** of Inactive, Play Once, Loop, and other responses are presented. When Mouse State Fire is chosen, Object Selected Audio Setting **309b4** is replaced with FIRE Audi Setting, with appropriate choices presented.

[0144] When Work with Child Objects and Mouse Overs button **309b5** is selected, a Child Object Mode box pops up, allowing a user to create a child object with shortcut to Menu bar **301** actions that may be used define child objects.

[0145] Figure 3D illustrates one embodiment of an Animation Tab **309c**, which includes all animations and timelines. The Color Tab includes all the possible color attributes, which may vary significantly by object type.

## PCT/US2009/039695

[0146] Animation Tab **309c** includes settings involved in animation and timelines that may be associated with objects. One embodiment of Animation Tab **309c** is shown, without limitation, in Figure 3D, and is described, in Rempell ("Rempell").

[0147] A Color Tab **309d** includes a color palette for selecting object colors.

[0148] Bindings Tab **309e** is where web component operations are defined and dynamic binding settings are assigned. Thus, for example, a UI object is selected from canvas **305**, and a web component may be selected and configured from the bindings tab. When the user's work is saved, binding information is associated with the UI object that will appear on screen **137**.

[0149] Figure 3E illustrates one embodiment of Bindings Tab and includes, without
limitation, the following portions: Web Component and Web Services Operations 309e1,
Attributes Exposed list 309e2, panel 309e3 which includes dynamic binding of server-side data
base values to attributes for the selected object, Default Attribute Value 309e4, Database Name
309e5, Table Name 309e6, Field Name 309e7, Channel Name 309e8, Channel Feed 309e9,
Operation 309e10, Select Link 309e11, and Link Set checkbox 309e12.

[0150] Web Component and Web Services Operations **309e1** includes web components that may be added, edited or removed from a selected object. Since multiple web components can be added to the same object, any combination of mash-ups of 3rd party web services is possible. When the "Add" button of Web Component and Web Services Operations **309e1** is selected, a pop-up menu **319**, as shown in Figure 3F, appears on publisher interface **300**. Pop-up menu **319** includes, but is not limited to, the options of: Select a Web Component **319a**; Select Results Page **319b**; Activation Options **319c**; Generate UI Objects **319d**; and Share Web Component **319e**.

[0151] The Select a Web Component **319a** portion presents a list of web components. As discussed herein, the web components are registered and are obtained from web component registry **220**.

[0152] Select Results Page **319b** is used to have the input and output on different pages - that is, when the Results page is different from Input page. The default selected results page is either the current page, or, if there are both inputs and outputs, it will be set provisionally to the next page in the current page order, if one exists.

[0153] Activation Options **319c** include, if there are no Input UI Objects, a choice to either "Preload" the web component, similar to how dynamic binding, or have the web component executed when the "Results" page is viewed by the consumer.

#### PCT/US2009/039695

[0154] Generate UI Objects **319c**, if selected, will automatically generate the UI objects. If not selected, then the author will bind the Web Component Inputs and Results to previously created UI Objects.

[0155] Share Web Component **319e** is available and will become selected under the following conditions: 1) Web Component is Selected which already has been used by the current Application; or 2) the current Input page is also a "Result" page for that Web component. This permits the user of device **130**, after viewing the results, to extend the Web Component allowing the user to make additional queries against the same Web Component. Examples of this include, but are not limited to, interactive panning and zooming for a Mapping Application, or additional and or refined searches for a Search Application.

[0156] Dynamic Binding permits the binding of real time data, that could either reside in a 3<sup>rd</sup> party server-side data base, or in the database maintained by Feed Collector **1010** for aggregating live RSS feeds, as described subsequently with reference to FIG. 10.

[0157] Referring again to FIG. 3E, Attributes Exposed list **309e2** are the attributes available for the selected object that can be defined in real time through dynamic binding.

[0158] Panel **309e3** exposes all the fields and tables associated with registered server-side data bases. In one embodiment, the user would select an attribute from the "Attributes Exposed List" and then select a data base, table and field to define the real time binding process. The final step is to define the record. If the Feed Collector data base is selected, for example, then the RSS "Channel Name" and the "Channel Feed" drop down menus will be available for symbolically selected the record. For other data bases the RSS "Channel Name" and the "Channel Feed" drop down menus are replaced by a "Record ID" text field.

[0159] Default Attribute Value **309e4** indicates the currently defined value for the selected attribute. It will be overridden in real time based on the dynamic linkage setting.

[0160] Database Name **309e5** indicates which server side data base is currently selected.

[0161] Table Name **309e6** indicates which table of the server side data base is currently selected.

[0162] Field Name **309e7**, indicates which field form the selected table of the server side data base is currently selected.

## PCT/US2009/039695

[0163] Channel Name **309e8** indicates a list of all the RSS feeds currently supported by the Feed Collector. This may be replaced by "Record ID" if a data base other than the Feed Collector **1010** is selected.

[0164] Channel Feed **309e9** indicates the particular RSS feed for the selected RSS Channel. Feed Collector **1010** may maintain multiple feeds for each RSS channel.

[0165] Operation **309e10**, as a default operation, replaces the default attribute value with the real time value. In other embodiments this operation could be append, add, subtract, multiply or divide.

[0166] Select Link **309e11** a button that, when pressed, creates the dynamic binding. Touching the "Select Link" will cause the current data base selections to begin the blink is some manner, and the "Select Link" will change to "Create Link". The user could still change the data base and attribute choices. Touching the "Create Link" will set the "Link Set" checkbox and the "Create Link" will be replaced by "Delete Link" if the user wishes to subsequently remove the link. When the application is saved, the current active links are used to create the SPDL.

[0167] Link Set checkbox **309e12** indicates that a link is currently active.

[0168] An example of the design of a display is shown in Figures 4A and 4B according the system 100, where FIG. 4A shows publisher interface 300 having a layout 410 on canvas 305, and FIG. 4B shows a device 130 having the resulting layout 420 on screen 137. Thus, for example, authoring platform 110 is used to design layout 410. Authoring platform 110 then generates an Application and a Player specific to device 130 of FIG. 4B. The Application and Player are thus used by device 130 to produce layout 420 on screen 137.

[0169] As illustrated in FIG. 4A, a user has placed the following on canvas **305** to generate layout **410**: text and background designs **411**, a first text input box **413**, a second text input box **415**, and a button **417**. As an example which is not meant to limit the scope of the present invention, layout **410** is screen prompts a user to enter a user name in box **413** and a password in box **415**, and enter the information by clicking on button **417**.

[0170] In one embodiment, all UI objects are initially rendered as Java objects on canvas**305**. When the Application is saved, the UI objects are transformed into the PDL, as described subsequently.

## PCT/US2009/039695

[0171] Thus, for example, layout **410** may be produced by the user of authoring platform **110** selecting and placing a first Text Field as box **413** then using the Resource Inspector **309** portion of interface **300** to define its attributes.

# **DEVICE USER EXPERIENCE**

[0172] Systems **100** and **200** provide the ability for a very large number of different types of user experiences. Some of these are a direct result of the ability of authoring platform **110** to bind UI objects to components of web services. The following description is illustrative of some of the many types of experiences of using a device **130** as part of system **100** or **200**.

[0173] Device **130** may have a one or more of a very powerful and broad set of extensible navigation objects, as well as object- and pointer-navigation options to make it easy to provide a small mobile device screen **137** with content and to navigate easily among page views, between Applications, or within objects in a single page view of an Application.

[0174] Navigation objects include various types of launch strips, various intelligent and userfriendly text fields and scrolling text boxes, powerful graphical complex lists, as well as Desktoplevel business forms. In fact, every type of object can be used for navigation by assigning a navigation event to it. The authoring tool offers a list of navigation object templates, which then can be modified in numerous ways.

# Launch Strips and Graphical List Templates Launch Strips

[0175] Launch strips may be designed by the user of authoring platform **110** with almost no restrictions. They can be stationary or appear on command from any edge of the device, their size, style, audio feedback, and animations can be freely defined to create highly compelling experiences.

[0176] Figure 5 shows a display **500** of launch strips which may be on displayed canvas **305** or on screen **137** of device **130** having the proper Player and Application. Display **501** includes a portal-type Launch Strip **501** and a channel-type Launch Strip **502**, either one of which may be included for navigating the Application.

[0177] Launch Strip 501 includes UI objects 501a, 501b, 501c, 501d, and 501e that that
becomes visible from the left edge of the display, when requested. UI objects 501a, 501b, 501c,
501d, and 501e are each associated, through resource inspector 309 with navigational
instructions, including but not limited to navigating to a different Applications Page, or

### PCT/US2009/039695

requesting web content. When the Applications Page, having been saved by authoring platform **110** and transferred to display **130**, is executed on device **130**, a user of the device may easily navigate the Application.

[0178] Launch Strip **502** includes UI objects **502b**, **502c**, **502d**, and **503e** that that becomes visible from the bottom of the display, when requested. UI objects **501a**, **501b**, **501c**, **501d**, and **501e** are each associated, through resource inspector **309** with navigational instructions, including but not limited to navigating to a different Applications Page, or requesting web content. Launch Strip **502** also includes UI objects **502a** and **503g**, which include the graphic of arrows, and which provide access to additional navigation objects (not shown) when selected by a user of device **130**. Launch strip **502** may also include sound effects for each channel when being selected, as well as popup bubble help.

[0179] Additional navigational features are illustrated in Figure 6A as a display of a Channel Selection List **601a**, in Figure 6B as a display of a Widget Selection List **601b**, and in Figure 6C as display of a Phone List **601c**. Lists **601a**, **601b**, and **601c** may be displayed on canvas **305** or on screen **137** of device **130** having the proper Player and Application. As illustrated, graphical lists **601a**, **601b**, and **601c** may contain items with many possible text and image elements. Each element can be defined at authoring time and/or populated dynamically through one or more Web Service **250** or API. Assignable Navigation Events. All objects, and/or all elements within an object, can be assigned navigation events that can be extended to registered web services or APIs. For example, a Rolodex-type of navigation event can dynamically set the starting slide of the targeted page view (or the starting view of a targeted Application).

[0180] In the embodiment of FIGS. 6A, 6B, and 6C, each list **601a**, **601b**, and **601c** has several individual entries that are each linked to specific actions. Thus Channel Selection List **601a** shows three objects, each dynamically linked to a web service (ESPN, SF 49ers, and Netflix) each providing a link to purchase or obtain items from the Internet. Widget Selection List **601b** includes several objects presenting different widgets for selecting. Phone List **601c** includes a list phone number objects of names that, when selected by a user of device **130** cause the number to be dialed Entries in Phone List **601c** may be generated automatically from either the user's contact list that is resident on the device, or though a dynamic link to any of user's chosen server-side facilities such as Microsoft Outlook, Google Mail, etc. In one embodiment, Phone List **601c** may be generated automatically using a web component assigned to the Application, which would automatically perform those functions.

### PCT/US2009/039695

[0181] In another embodiment, authoring platform **110** allows a navigation selection of objects with a Joy Stick and/or Cursor Keys in all 4 directions. When within a complex object the navigation system automatically adopts to the navigation needs for that object. For coordinate sensitive objects such as geographical information services (GIS) and location-based services (LBS) or virtual tours a soft cursor appears. For Lists, scrolling text areas and chats, Launch strips, and slide shows the navigation process permits intuitive selection of elements within the object. Scroll bars and elevators are optionally available for feedback. If the device has a pointing mechanism then scroll bars are active and simulate the desktop experience.

# Personalization and Temporal Adoption

[0182] System **100** and **200** permit for the personalization of device **130** by a variety of means. Specifically, what is displayed on screen **137** may depend on either adoption or customization. Adoption refers to the selection of choices, navigation options, etc. are based on user usage patterns. Temporal adoption permits the skins, choices, layouts, content. widgets, etc. to be further influenced by location (for example home, work or traveling) and time of day (including season and day of week). Customization refers to user selectable skins, choices, layouts, dynamic content, widgets, etc. that are available either through a customization on the phone or one that is on the desktop but dynamically linked to the user's other internet connected devices.

[0183] To support many personalization functions there must be a convenient method for maintaining, both within a user's session, and between sessions, memory about various user choices and events. Both utilizing a persistent storage mechanism on the device, or a database for user profiles on a server, may be employed.

[0184] Figure 7 shows a display 700 of a mash-up which may be on displayed canvas 305 or on screen 137 of device 130 having the proper Player and Application. Display 700 includes several object 701 that have been dynamically bound, including an indication of time 701a, an indication of unread text messages 701b, an RSS news feed 701c (including 2 "ESPN Top Stories" 701c1 and 701c2), components 701d from two Web Services - a weather report ("The Weather Channel"), and a traffic report 701e ("TRAFFIC.COM").

[0185] In assembling the information of display 700, device 130 is aware of the time and location of the device - in this example the display is for a workday when a user wakes. Device 130 has been customized so that on a work day morning the user wishes to receive the displayed information. Thus in the morning, any messages received overnight would be flagged, the user's

### PCT/US2009/039695

favorite RSS sports feeds would be visible, today's weather forecast would be available, and the current traffic conditions between the user's home and office would be graphically depicted. User personalization settings may be maintained as persistent storage on device **130** when appropriate, or in a user profile which is maintained and updated in real-time in a server-side data base.

## PUSH CAPABLE SYSTEMS

[0186] In another embodiment system **100** or **200** is a push-capable system. As an example, of such systems, short codes may be applied to cereal boxes and beverage containers, and SMS text fields can be applied to promotional websites. In either case, a user of device **130** can text the short code or text field to an SMS server, which then serves the appropriate Application link back to device **130**.

[0187] Figure 8 is a schematic of an embodiment of a push enabled system 800. System 800 is generally similar to system 100 or 200. Device 130 is shown as part of a schematic of a push capable system 800 in Figure 8. System 800 includes a website system 801 hosting a website 801, a server 803 and a content server 805. System 801 is connected to servers 803 and/or 805 through the Internet. Server 803 is generally similar to server 120, servers 805 is generally similar to server 140.

[0188] In one embodiment, a user sets up a weekly SMS update from website system **801**. System **801** provides user information to server **803**, which is an SMS server, when an update is ready for delivery. Server **803** provides device **130** with an SMS indication that the subscribed information is available and queries the user to see if they wish to receive the update. Website **801** also provides content server **805** with the content of the update. When a user of device **130** responds to the SMS query, the response is provided to content server **805**, which provides device **130** with updates including the subscribed content.

[0189] In an alternative embodiment of system 800, server 803 broadcasts alerts to one or more devices 130, such as a logical group of devices. The user is notified in real-time of the pending alert, and can view and interact with the massage without interrupting the current Application.

[0190] Figure 9 is a schematic of an alternative embodiment of a push enabled system **900**. System **900** is generally similar to system **100**, **200**, or **800**. In system **900** a user requests information using an SMS code, which is delivered to device **130**. System **900** includes a promotional code **901**, a third-party server **903**, and content server **805**. Server **803** is connected to servers **803** and/or **805** through the Internet, and is generally similar to server **120**.

## PCT/US2009/039695

[0191] A promotional code 901 is provided to a user of device 130, for example and without limitation, on print media, such as on a cereal box. The use of device 130 sends the code server 903. Server 903 then notifies server 805 to provide certain information to device 130. Server 805 then provides device 130 with the requested information.

# **DEVICE ROUTINES**

[0192] Device routines **114** may include, but are not limited to: an authoring tool SDK for custom code development including full set of Java APIs to make it easy to add extensions and functionality to mobile Applications and tie Applications to back-end databases through the content server **140**; an expanding set of web services **250** available through the authoring tool SDK; a web services interface to SOAP/XML enabled web services; and an RSS/Atom and RDF feed collector **1010** and content gateway **1130**.

# Authoring tool SDK for custom code development including full set of Java APIs

[0193] In one embodiment, authoring platform **110** SDK is compatible for working with various integrated development environments (IDE) and popular plug ins such as J2ME Polish. In one embodiment the SDK would be another plug in to these IDEs. A large and powerful set of APIs and interfaces are thus available through the SDK to permit the seamless extension of any Application to back end business logic, web services, etc. These interfaces and APIs may also support listeners and player-side object operations.

[0194] There is a large set of listeners that expose both player-side events and dynamically linked server side data base events. Some examples of player side events are: player-side time based event, a page entry event, player-side user interactions and player-side object status. Examples of server-side data base events are when a particular set of linked data base field values change, or some filed value exceeds a certain limit, etc.

[0195] A superset of all authoring tool functionality is available through APIs for layer-side object operations. These include, but are not limited to: page view level APIs for inserting, replacing, and or modifying any page object; Object Level APIs for modifying any attribute of existing objects, adding definitions to attributes, and adding, hiding or replacing any object.

# Authoring tool SDK available Web Services

[0196] The APIs permit, without limit, respond, with or without relying on back-end business logic, that is, logic that what an enterprise has developed for their business, to any player-side

event or server-side dynamically linked data-base, incorporating any open 3rd party web service(s) into the response.

# RSS/ATOM AND RDF FEED CONVERSION WEB SERVICE

[0197] Figure 10 is a schematic of one embodiment a system 1000 having a feed collector 1010. System 1000 is generally similar to system 100, 200, 800, or 900. Feed collector 1010 is a server side component of system 100 that collects RSS, ATOM and RDF format feeds from various sources 1001 and aggregates them into a database 1022 for use by the Applications built using authoring platform 110.

[0198] Feed collector **1010** is a standard XML DOM data extraction process, and includes Atom Populator Rule **1012**, RSS Populator Rule **1013**, RDF Populator Rule **1014**, and Custom Populator Rule **1016**, DOM XML Parsers **1011**, **1015**, and **1017**, Feed Processed Data Writer **1018**, Custom Rule Based Field Extraction **1019**, Rule-based Field Extraction **1020**, Channel Data Controller **1021**, and Database **1022**.

[0199] The feed collector is primarily driven by two sets of parameters: one is the database schema (written as SQL DDL) which defines the tables in the database, as well as parameters for each of the feeds to be examined. The other is the feed collection rules, written in XML, which can be used to customize the information that is extracted from the feeds. Each of the feeds is collected at intervals specified by the feed parameter set in the SQL DDL.

[0200] Feed collector **1010** accepts information from ATOM, RDF or RSS feed sources **1001.** Using a rules-based populator, any of these feeds can be logically parsed, with any type of data extraction methodology, either by using supplied rules, or by the author defining their own custom extraction rule. The rules are used by the parser to parse from the feed sources, and the custom rule base field extraction replaces the default rules and assembles the parsed information into the database

[0201] In particular, Atom Populator Rule **1012**, RSS Populator Rule **1013**, RDF Populator Rule **1014**, Custom Populator Rule **1016**, and DOM XML Parsers **1011**, **1015**, and **1017** are parse information from the feeds **1001**, and Feed Processed Data Writer **1018**, Custom Rule Based Field Extraction **1019**, Rule-based Field Extraction **1020**, and Channel Data Controller **1021**, supply the content of the feeds in Database **1022**, which is accessible through content server **140**.

### PCT/US2009/039695

[0202] Figure 11 is a schematic of an embodiment of a system **1100** having a Mobile Content Gateway **1130**. System **1100** is generally similar to system **100**, **200**, **800**, **900**, or **1000**. System **1100** includes an SDK **1131**, feed collector **1010**, database listener **1133**, transaction server **1134**, custom code **1135** generated from the SDK, Java APIs, Web Services **1137**, and PDL snippets compacted objects **1139**. System **1100** accepts input from Back End Java Code Developer **1120** and SOAP XML from Web Services **1110**, and provides dynamic content to server **140** and Players to devices **130**.

[0203] In one embodiment authoring platform **110** produces a Server-side PDL (SPDL) at authoring time. The SPDL resides in server **120** and provides a logical link between the Application's UI attributes and dynamic content in database **1022**. When a user of device **130** requests dynamic information, server **120** uses the SPDL to determine the link required to access the requested content.

[0204] In another embodiment Web Services **1137** interface directly with 3rd party Web Services **1110**, using SOAP, REST, JAVA, JavaScript, or any other interface for dynamically updating the attributes of the Application's UI objects.

# XSP WEB PAGES AS A WEB SERVICE

[0205] In one embodiment, a PDL for a page is embedded within an HTML shell, forming one XSP page. The process of forming XSP includes compressing the description of the page and then embedding the page within an HTML shell.

[0206] In another embodiment, a PDL, which contains many individual page definitions, is split into separate library objects on the server, so that each page can to presented as a PDL as part of a Web Service.

[0207] Prior to compression the code has already been transformed so that there are no dependencies on the original programming language (Java), and The code and data have been reduced by 4 to 10 times.

[0208] Compression has two distinct phases. The first takes advantage of how the primitive representations had been assembled, while the second utilizes standard LZ encoding.

[0209] The final result is an overall reduction of 40 to 100 times the original size as represented by Java serialized objects.

### PCT/US2009/039695

[0210] One embodiment for compacting data that may be used is described in Rempell. In that patent the compressed data is described as being a database. The terminology used here is a PDL, that is the "internal database" of Rempell is equivalent to the PDL of the present Application.

[0211] In Rempell, a process for compacting a "database" (that is, generating a compact PDL) is described, wherein data objects, including but not limited to, multi media objects such as colors, fonts, images, sound clips, URLs, threads, and video, including multi level animation, transformation, and time line are compacted. As an extension to Rempell in all cases these objects are reduced and transformed to Boolean, integer and string arrays.

[0212] The compression technique involves storing data in the smallest arrays necessary to compactly store web page information. The technique also includes an advanced form of delta compression that reduces integers so that they can be stored in a single byte, a shigh water marks.

[0213] Thus, for example, the high water mark for different types of data comprising specific web site settings are stored in a header record as Boolean and integer variables and URL and color objects. Data that defines web page, paragraph, text button, and image style and text button, image and paragraph high watermark settings can be stored in one-dimensional arrays as Boolean, integer and string variables and URL, font, image or thread objects at. The URL, color, font, image and thread objects can also be created as required

[0214] Data that defines text button, image, paragraph, or other parent objects and paragraph line high watermark settings can be stored in two-dimensional arrays (by web page and by object number) as Boolean, integer, string, floating point variables and URLs. Again, the URL, color, font, image, audio clip, video clip, text area and thread objects can also be created as required. Data that defines a paragraph line and paragraph line segment high watermarks can be stored in three-dimensional arrays (by web page, by paragraph number, and by line number) as Boolean, integer or string variables. Again, the URL, color or font objects can be created as required. Data that defines a paragraph line segment can be stored into four-dimensional arrays (by web page, by paragraph number, by line number and by line number segment) as Boolean, integer or string variables or URL, color and font objects.

[0215] As a data field is added, changed or deleted, a determination is made at on whether a value for a given high watermark needs to be changed. If so, it is updated. As a specific method in the build engine is called, a determination is made on whether a feature flag needs to be set.

## PCT/US2009/039695

For example, if a particular JAVA method is called, which requires an instance of a certain JAVA Class to be executed by the run time engine, then that JAVA Class is flagged, as well as any supporting methods, variables and/or object definitions.

[0216] In one implementation, the header record, the style record, the web page record, and the object records, are carefully defined in a specific order, written in that order, and explicitly cast by object type when read by the run time engine. Exception handling can be implemented to recover from any errors. This helps assure that data integrity is maintained throughout the build and run time processes.

[0217] Also described in Rempell is the "run generation process." This is equivalent generating a Player in the present application. This process starts when the build process detects that the user is finished defining the web site (user has saved the web site and invokes the run generation process), and concludes with the actual uploading of all the necessary web site run time files to the user's server.

[0218] In one embodiment, the PDL includes a first record, a "Header" record, which contains can include the following information:

[0219] 1: A file format version number, used for upgrading database in future releases.

[0220] 2: The default screen resolution, in virtual pixels, for both the screen width and height. This is usually set to the web designer's screen resolution, unless overwritten by the user.

[0221] 3: Whether the Application is a web site.

[0222] 4: Virtual web page size settings. A calculation is performed by the build engine method, in order to calculate what the maximum web page length is, after reformatting all paragraphs on all internal web pages, based on the default screen resolution.

[0223] 5: Web page and styles high watermarks.

[0224] 6: The Websitename.

[0225] As new web pages or new objects are created by the user, or as text is added to or deleted from a paragraph, or as new styles are created or deleted, appropriate high watermarks are set, in order to show the current number of each of these entities. Thus, the values for the number of active web pages and the number of text button, image, paragraph or other styles are written as high watermarks in the header. The high watermarks for the number of text button,

#### PCT/US2009/039695

image, paragraph or other objects that exist for each web page, the number of lines for each paragraph object, and the number of line segments for each paragraph line are written within the body of the PDL, and used as settings for each of the loops in the four-dimensional data structure. Because no structural limits are set on the number of web pages, objects per web page, styles, or paragraph size, these high watermarks greatly reduce the external database file size, and the time it takes for the run time engine to process the data stored in its database.

[0226] The settings for all paragraph, text button and image styles are then written as a style record based on their high watermark. This data includes Boolean and integer variables, and font and color objects, written as a one-dimensional array, based on the high watermark values for the number of styles that exist.

The body of the PDL is then written. All Boolean values are written inside a four-[0227] dimensional loop. The outside loop contains the Boolean values used to define web pages (i.e. a one-dimensional array definition) as well as the high watermarks for the number of text button, image, paragraph or other objects per web page, with the loop set at the high watermark which defines the number of existing web pages for this web site structure. The second level consists of three or more two dimensional loops with the loops set to the high watermarks defining the actual number of text button, image, and paragraph or other objects that appear on any given web page and contains the values used to define web page objects ((i.e. a two-dimensional array definition; web page number by object number). Included within the loop for paragraph objects are the high watermarks for the number of lines for each paragraph object. The third loop is set by the high watermark defining the actual number of paragraph lines that for all paragraphs on any web page and contains the values used to define paragraph lines (i.e. a three-dimensional array definition; web page number by object number by paragraph line.) Included within the loop for paragraph lines are the high watermarks for the number of line segments for each paragraph line. The inner most loop is set by the high watermarks defining the number of line segments per paragraph line and contains the values used to define paragraph line segments (i.e. a four-dimensional array definition; web page number by object number by paragraph line by paragraph line segment).

[0228] All integer values are written inside a four-dimensional loop. Their four loops are controlled by the same high watermark settings as used for the Boolean records, and they describe the same logical entities.

[0229] Multimedia objects are written inside a two-dimensional loop. They include URL, color, and font objects, and can include other types of objects. A URL object is the encoded form of a URL Address, used by a web browser or a JAVA method to access files and web addresses.
#### PCT/US2009/039695

All multimedia objects must be serialized before they can be written. This means that the objects are converted into a common external definition format that can be understood by the appropriate descrialization technique when they are read back in and cast into their original object structure. The outside loop contains web page related objects, and the inner loop contains image, text button, paragraph, etc. related URL, color, and font objects. The outer loop is defined by the web page high watermark and the inner loops by the high watermarks for the actual number of text button, image, paragraph or other objects on a web page.

[0230] String records are written inside a four-dimensional loop. The outer loop may be empty. The second loop can include the string values for text button objects, audio and video filenames, and audio and video channel names. The third loop contains values for paragraph line related data, and the innermost loop contains the values for paragraph line segment definitions. The string records are controlled by the same high watermarks as those used for Boolean and integer records. String records are stored utilizing an appropriate field delimiter technology. In one implementation, a UTF encoding technology that is supported by JAVA is utilized.

[0231] Single and double floating-point, and long integer records are written inside a twodimensional loop. The outer loop may be empty. The inner loop contains mathematical values required for certain animations and image processing algorithms. The single and double floatingpoint, and long integer records are controlled by the same high watermarks as those used for Boolean and integer records.

[0232] In one embodiment, a versionizing program analyzes the feature flags, and only those variable definitions, defined in the "Main" object class, relating to the object classes and methods that will be executed at run time, are extracted. All references to object classes that will be called at run time are extracted, creating the source code for the run engine "Main" object class that is ready for compilation.

[0233] All external image, video and audio files are resolved. The external references can be copied to designated directories, either on the user's local disk or file server. The file Pathnames can be changed to reflect these new locations. During the installation of the build tools, the necessary class libraries are either installed on the local system or made available on the server where the build tools can be optionally located. The necessary environmental variables are set to permit normal access to the required class libraries.

[0234] The customized run engine and a library of the referenced run time classes are compiled and converted into byte code. Finally, the run time engine for the web site is created.

#### PCT/US2009/039695

The required set of class objects required at run time is flagged for inclusion into the CAB/JAR file.

[0235] Next, an HTML Shell File (HSF) is constructed. The first step of this process is to determine whether the dynamic web page and object resizing is desired by testing the Application setting. If the Application was a web page, and thus requiring dynamic web page and object resizing, virtual screen resolution settings are placed in an appropriate HTML compliant string. If the Application is a banner or other customized Application, the absolute values for the run time object (applet size) height and width are placed in an appropriate HTML compliant string as absolute width and height values.

[0236] An analysis is made for the background definition for the first internal web page. If a background pattern is defined, an appropriate HTML compliant string for setting the HTML "background" to the same background image is generated. If the first web page definition is a color instead, then the RGB values from those colors are converted to hexadecimal and an appropriate HTML compliant String is generated setting the "bgcolor" to the required hexadecimal value. This process synchronizes the web page background with the background that will be drawn by the web browser when it first interprets the HSF.

[0237] Thereafter, a JAVA method generates HTML and JavaScript compliant strings, that when executed by a web browser, generate additional sets of HTML and JavaScript compliant strings that are again executed by the web browser. More specifically, if the Application required dynamic web page and object resizing then JavaScript and HTML compliant strings are generated so that, when interpreted by the web browser at the time the HTML Shell File is initialized, the screen resolution sensing JAVA applet (SRS) will be executed. JavaScript code is generated in order to enable JavaScript to SRS applet communication. In one implementation, the code is generated by performing the following functions:

[0238] 1: Determine the current web browser type.

[0239] 2: Load the SRS from either a JAR or CAB File, based on web browser type.

[0240] 3: Enter a timing loop, interrogating when the SRS is loaded.

[0241] 4: When the SRS returns an "available" status, interrogate the SRS, which will return the current screen and window's actual height and width.

#### PCT/US2009/039695

[0242] 5: Convert the virtual screen resolution settings into appropriate absolute screen width and height values.

[0243] Strings defining additional JavaScript code are generated that perform the following steps at the time the HSF is initialized by the web browser:

[0244] 1: Generate HTML compliant strings that set the run time engine's applet size to the appropriate values.

[0245] 2: Generate an HTML complaint string that contains a "param" definition for linking the run time engine to the PDL.

[0246] 3: Generate an HTML complaint string, dependent upon the type of web browser, which causes the current web browser to load either the JAR or the CAB File(s).

[0247] 4: Generate JavaScript Code compliant strings that create and dynamically write the applet size defining HTML strings utilizing the JavaScript "document.write" function. This dynamically created code causes the web browser to execute the run time engine, in the correctly sized window, from the correct JAR or CAB file, and linked to the external database.

[0248] The writing out the above-generated HTML and JavaScript compliant strings creates the HSF. The necessary security policy permissions are asserted, and a "Websitename".html file is created.

[0249] In one embodiment, the processes for creating the CAB and JAR Files is as follows. The image objects, if any, which were defined on the first internal web page are analyzed. If they are set to draw immediately upon the loading of the first web page, then they are flagged for compression and inclusion in the CAB and JAR Files. The feature flags are analyzed to determine which JAVA classes have been compiled. These class files are flagged for compression and inclusion in the library CAB and JAR Files. Strings that are BAT compliant definitions are created that will, when executed in DOS, create compressed CAB and JAR Files. These CAB and JAR Files contain the compressed versions of all necessary JAVA class files, image files, the "Websitename".class, customized run time engine file, and the "Websitename".dta database file. In one implementation of the invention, two BAT files are created. The first, when executed, will create a CAB/JAR file with the "Websitename".dta database file and the customized "main" run time engine, excluding all the image and button object animation, transformation, and image processing code. The second BAT file, when

#### PCT/US2009/039695

executed, will create a CAB/JAR file with all the library of all the referenced image and button object animation, transformation, and image processing code.

[0250] The necessary security policy permissions for file creation are then asserted, and "Websitename".bat and "Websitename".bat and "Websitename".bat files are then executed under DOS, creating compressed "Websitename".cab and "Websitenamelib".cab files and compressed "Websitename" jar and "Websitenamelib" jar files. The HTML Shell File and the JAR and CAB files are then, either as an automatic process, or manually, uploaded to the user's web site. This completes the production of an XSP page that may be accessed through a web browser.

#### DISPLAYING CONTENT ON A DEVICE

#### **Decompression Management**

[0251] Authoring platform **110** uses compaction to transform the code and data in an intelligent way while preserving all of the original classes, methods and attributes. This requires both an intelligent server engine and client (handset) Player, both of which fully understand what the data means and how it will be used.

[0252] The compaction technology described above includes transformation algorithms that deconstruct the logic and data into their most primitive representations, and then reassembles them in a way that can be optimally digested by further compression processing. This reassembled set of primitive representations defines the PDL of authoring platform **110**.

[0253] Prior to compression the code has already been transformed so that there are no dependencies on the original programming language (Java). The data is then compressed by first taking advantage of how the primitive representations had been assembled, and then by utilizing standard LZ encoding. The final result is an overall reduction of 40 to 100 times the original size as represented by Java serialized objects.

[0254] The Player, when preparing a page view for execution, decompresses and then regenerate the original objects, but this time in compliance with the programming APIs of device **130**. Specifically, device **130** operates on compacted image pages, one at a time. The cache manager retrieves, decompresses, and reassembles the compacted page images into device objects, which are then interpreted by device **130** for display on screen **137**.

## WO 2009/126591 Response Director

[0255] In one embodiment, system **100** includes a Response Director, which determines a user's handset, fetches the correct Application from different databases, and delivers a respective highly compressed Application in a PDL format over the air (OTA).

[0256] In one embodiment, the Response Director operates on a network connected computer to provide the correct Player to a given device based on the information the device sent to it. As an example, this may occur when a device user enters their phone number into some call-to-action web page. The response director is called and sends an SMS message to the device, which responds, beginning the recognition process.

[0257] Figure 12 illustrates one embodiment of a system 1200 that includes a response director 210, a user agent database 1201, an IP address database 1203, and a file database 1205. System 1200 is generally similar to system 100, 200, 800, 900, 1000, or 1100.

[0258] Databases **1201**, **1203**, and **1205** may reside on server **120**, **210**, or any computer system in communication with response director **210**. System **1200**, any mobile device can be serviced, and the most appropriate Application for the device will be delivered to the device, based on the characteristics of the device.

[0259] User agent database 1201 includes user agent information regarding individual devices 130 that are used to identify the operating system on the device. IP address database
1203 identifies the carrier/operator of each device 130. File database 1205 includes data files that may operate on each device 130.

[0260] The following is an illustrative example of the operation of response director **210**. First, a device **1300** generates an SMS message, which automatically sends an http:// stream that includes handset information and its phone number to response director **210**. Response director **210** then looks at a field in the http header (which includes the user agent and IP address) that identifies the web browser (i.e., the "User Agent"). The User Agent prompts a database lookup in user agent database **1201** which returns data including, but not limited to, make, model, attributes, MIDP 1.0 MIDP 2.0, WAP and distinguishes the same models from different countries. A lookup of the IP address in IP address **1203** identifies the carrier/operator.

[0261] File database **1205** contains data types, which may include as jad1, jad2, html, wml/wap2, or other data types, appropriate for each device **130**. A list of available Applications are returned to a decision tree, which then returns, to device **130**, the Application that is

#### PCT/US2009/039695

appropriate for the respective device. For each file type, there is an attributes list (e.g., streaming video, embedded video, streaming audio, etc.) to provide enough information to determine what to send to the handset.

[0262] Response director **210** generates or updates an html or jad file populating this text file with the necessary device and network dependent parameters, including the Application dependent parameters, and then generate, for example, a CAB or JAD file which contains the necessary Player for that device. For example, the jad file could contain the operator or device type or extended device-specific functions that the player would then become aware of.

[0263] If there is an Application that has a data type that device **130** cannot support, for example, video, response director **210** sends an alternative Application to the handset, for example one that has a slide show instead. If the device cannot support a slide show, an Application might have text and images and display a message that indicates it does not support video.

[0264] Another powerful feature of response director **210** is its exposed API from the decision tree that permits the overriding of the default output of the decision tree by solution providers. These solution providers are often licensees who want to further refine the fulfillment of Applications and Players to specific devices beyond what the default algorithms provide. Solution providers may be given a choice of Applications and then can decide to use the defaults or force other Applications.

[0265] Authoring platform **110** automatically scales Applications at publishing time to various form factors to reduce the amount of fragmentation among devices, and the Response Director serves the appropriately scaled version to the device. For example, a QVGA Application will automatically scale to the QCIF form factor. This is important because one of the most visible forms of fragmentation resides in the various form factors of wireless, and particularly mobile, devices, which range from 128x128, 176x208, 240x260, 220x220, and many other customized sizes in between.

[0266] Figure 13 is a schematic of an embodiment of a system **1300**. System **1300** is generally similar to system **1200**. System **1300** is an overview of the entire Player fulfillment process, starting with the generation of players during the player build process.

#### PCT/US2009/039695

[0267] System 1300 includes response director 210, a device characteristics operator and local database 1301, a player profile database 1303 and a player build process 1305, which may be authoring platform 110.

[0268] As an example of system **1300**, when response director **210** receives an SMS message from device **130**, the response director identifies the device characteristics operator and locale from database **1301** and a Player URL from database **1303** and provides the appropriate Player to the device.

[0269] In another embodiment, Player **P** extend the power of response director **210** by adapting the Application to the resources and limitations of any particular device. Some of these areas of adaptation include the speed of the device's microprocessor, the presence of device resources such as cameras and touch screens. Another area of adaptation is directed to heap, record store and file system memory constraints. In one embodiment, the Player will automatically throttle down an animation to the frame rate that the device can handle so that the best possible user experience is preserved. Other extensions include device specific facilities such as location awareness, advanced touch screen interactions, push extensions, access to advanced phone facilities, and many others

#### Memory Management

[0270] In one embodiment, Player **P** includes a logical page virtual memory manager. This architecture requires no supporting hardware and works efficiently with constrained devices. All page view images, which could span multiple Applications, are placed in a table as highly compacted and compressed code. A typical page view will range from 500 bytes up to about 1,500 bytes. (See, for example, the Rempell patent) When rolled into the heap and instantiated this code increases to the more typical 50,000 up to 250,000 bytes. Additional alert pages may also be rolled into the heap and superimposed on the current page view. Any changes to any page currently downloaded are placed in a highly compact change vector for each page, and rolled out when the page is discarded. Note that whenever an Application is visited that had previously been placed in virtual memory the Server is interrogated to see if a more current version is available, and, if so, downloads it. This means that Application logic can be changed in real-time and the results immediately available to mobile devices.

[0271] To operate efficiently with the bandwidth constraints of mobile devices, authoring platform **110** may also utilize anticipatory streaming and multi-level caching. Anticipatory streaming includes multiple asynchronous threads and IO request queues. In this process, the

#### PCT/US2009/039695

current Application is scanned to determine if there is content that is likely to be required in asyet untouched page views. Anticipatory streaming also looks for mapping Applications, where the user may zoom or pan next so that map content is retrieved prior to the user requesting it. For mapping applications, anticipatory streaming downloads a map whose size is greater than the map portal size on the device and centered within the portal. Any pan operation will anticipatory stream a section of the map to extend the view in the direction of the pan while, as a lower priority, bring down the next and prior zoom levels for this new geography. Zooming will always anticipatory stream the next zoom level up and down.

[0272] Multi-level caching determines the handset's heap through an API, and also looks at the record store to see how much memory is resident. This content is placed in record store and/or the file system, and may, if there is available heap, also place the content there as well. Multi-level caching permits the management of memory such that mobile systems best use limited memory resources. Multi-level caching is a memory management system with results similar to embedding, without the overhead of instantiating the content. In other words, with multi-level caching, handset users get an "embedded" performance without the embedded download. Note that when content is flagged as cacheable and is placed in persistent storage, a digital rights management (DRM) solution will be used.

[0273] One embodiment of each of the methods described herein is in the form of a computer program that executes on a processing system. Thus, as will be appreciated by those skilled in the art, embodiments of the present invention may be embodied as a method, an apparatus such as a special purpose apparatus, an apparatus such as a data processing system, or a carrier medium, e.g., a computer program product. The carrier medium carries one or more computer readable code segments for controlling a processing system to implement a method. Accordingly, aspects of the present invention may take the form of a method, an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects. Furthermore, the present invention may take the form of carrier medium (e.g., a computer program product on a computer-readable storage medium) carrying computer-readable program code segments embodied in the medium. Any suitable computer readable medium may be used including a magnetic storage device such as a diskette or a hard disk, or an optical storage device such as a CD-ROM.

[0274] It will be understood that the steps of methods discussed are performed in one embodiment by an appropriate processor (or processors) of a processing (i.e., computer) system executing instructions (code segments) stored in storage. It will also be understood that the

#### PCT/US2009/039695

invention is not limited to any particular implementation or programming technique and that the invention may be implemented using any appropriate techniques for implementing the functionality described herein. The invention is not limited to any particular programming language or operating system. It should thus be appreciated that although the coding for programming devices has not be discussed in detail, the invention is not limited to a specific coding method. Furthermore, the invention is not limited to any one type of network architecture and method of encapsulation, and thus may be utilized in conjunction with one or a combination of other network architectures/protocols.

[0275] Reference throughout this specification to "one embodiment," "an embodiment," or "certain embodiments" means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, appearances of the phrases "in one embodiment," "in an embodiment," or "in certain embodiments" in various places throughout this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures or characteristics may be combined in any suitable manner, as would be apparent to one of ordinary skill in the art from this disclosure, in one or more embodiments.

[0276] Throughout this specification, the term "comprising" shall be synonymous with "including," "containing," or "characterized by," is inclusive or open-ended and does not exclude additional, unrecited elements or method steps. "Comprising" is a term of art which means that the named elements are essential, but other elements may be added and still form a construct within the scope of the statement. "Comprising" leaves open for the inclusion of unspecified ingredients even in major amounts.

[0277] Similarly, it should be appreciated that in the above description of exemplary embodiments, various features of the invention are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure and aiding in the understanding of one or more of the various inventive aspects. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed invention requires more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed embodiment, and the invention may include any of the different combinations embodied herein. Thus, the following claims are hereby expressly incorporated into this Mode(s) for Carrying Out the Invention, with each claim standing on its own as a separate embodiment of this invention.

#### PCT/US2009/039695

[0278] Thus, while there has been described what is believed to be the preferred embodiments of the invention, those skilled in the art will recognize that other and further modifications may be made thereto without departing from the spirit of the invention, and it is intended to claim all such changes and modifications as fall within the scope of the invention. For example, any formulas given above are merely representative of procedures that may be used. Functionality may be added or deleted from the block diagrams and operations may be interchanged among functional blocks. Steps may be added or deleted to methods described within the scope of the present invention.

PCT/US2009/039695

## **WO 2009/126591** We Claim:

1. A system for generating code to provide content on a display of a platform, said system comprising:

a database of web services obtainable over a network;

an authoring tool configured to:

define an object for presentation on the display,

select a component of a web service included in said database,

associate said object with said selected component, and

produce code that, when executed on the platform, provides said selected component on the display of the platform.

2. The system of Claim 1, where said database includes definitions of input and/or output related to said web service.

3. The system of Claim 1, where said component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

4. The system of Claim 1, where said object is an input field for a chat.

5. The system of Claim 1, where said object is an input field for a web service.

6. The system of Claim 1, where said object is an input field usable to obtain said component, where said input field includes a text field, a scrolling text box, a check box, a drop down-menu, a list menu, or a submit button.

7. The system of Claim 1, where said component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.

8. The system of Claim 1, where said authoring tool is further configured to:

define a phone field or list; and

generate code that, when executed on the platform, allows a user to supply a phone number to said phone field or list.

9. The system of Claim 1, where said authoring tool is further configured to:

define a SMS field or list; and

generate code that, when executed on the platform, allows a user to supply an SMS address to said SMS field or list.

10. The system of Claim 1, where said code includes two or more codes, where one of said two or more codes is platform specific, and where one of said two or more codes in platform independent.

11. The system of Claim 1, where said code is provided over said network.

#### PCT/US2009/039695

12. A method for displaying content on a platform utilizing a database of web services obtainable over a network, said method comprising:

defining an object for presentation on the display; selecting a component of a web service included in said database; associating said object with said selected component; and

producing code that, when executed on the platform, provides said selected component on the display of the platform.

13. The method of Claim 12, where said database includes definitions of input and/or output related to said web service.

14. The method of Claim 12, where said component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

15. The method of Claim 12, where said object is an input field for a chat.

16. The method of Claim 12, where said object is an input field for a web service.

17. The method of Claim 12, where said object is an input field usable to obtain said component, where said input field includes a text field, a scrolling text box, a check box, a drop down-menu, a list menu, or a submit button.

18. The method of Claim 12, where said component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.

19. The method of Claim 12, further comprising:

defining a phone field or list; and

generating code that, when executed on the platform, allows a user to supply a phone number to said phone field or list.

20. The method of Claim 12, further comprising:

defining a SMS field or list; and

generating code that, when executed on the platform, allows a user to supply an SMS address to said SMS field or list.

21. The method of Claim 12, where said code includes two or more codes, where one of said two or more codes is platform specific, and where one of said two or more codes in platform independent.

22. The method of Claim 12, further comprising:

providing said code over said network.

 A method for providing information to a platform on a network, said method comprising: accepting a first code over the network, where said first code is platformdependent;

#### PCT/US2009/039695

providing a second code over the network, where said second code is platformindependent; and

executing said first code and said second code on the platform to provide web components obtained over the network.

24. The method of Claim 23, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

25. The method of Claim 23, where said component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.

26. The method of Claim 23, where said first code and said second code are generated using an authoring tool.

27. The method of Claim 23, where said first code is a Player.

28. The method of Claim 23, where said second code is an Application.

100 🔨



FIG. 1A



FIG. 1B



FIG. 2A

IPR2021-01146 Page 00160

3/18



FIG. 2B



5/18



309b Settings Animation. Bindings Events Color **Events and Web Services** None 309b1 2 Goto External Web Page replacing Current Frame Goto External Web Page Launched in a New Window Goto a specific Page View Goto External Web Page replacing the Top Frame Goto the next Page View Execute Java Script Method Pause/Resume Page Timeout Execute an Application Goto a Specific Slide in a Page View Exit Application Exit Player Place Phone Call Send String on FIRE Send String on FIRE or Numeric Keys 309b2 -309b3 **Advanced Interactive Settings Mouse State** Scroll Activation Enabled Selected Fire Timeline Entry Suppressed 309b4 **Object Selected Audio Settings** Enable Server Listener Inactive Submit Form Toggle Children on FIRE Hide non-related Children Select a Sound File 309b5 Work with Child Objects and Mouse Overs **Object Selected: Text Field** 

7/18

FIG. 3C

309с 🔍

Settin	ías I	Events	Animation	Color	Bindings
Activate Timeline	e Dek	bject Entry Tir Iy (Sec) Dir 1.0 1 No	neline Specifica ection Movem ne 👔 None	tions enf Duration( 2 2 .0	Sec) Frames
Spx Inactiv	ecifica R	itions for this	Object's Entry A	nimation Aud	io Track by Audio File
		<b>Object Anima</b>	ition Specificati	ons	
Delay( 1 2 3 4 5 6 7 8 9 10	Sec) .0 .2 .2 .3 .4 .5 .6 .7 .8 .9	Direction None Scroll Left Scroll Right Custom Multi-Point Seek Cursor Attach Deposit Send Home Carom NE	Movement <ul> <li>None</li> <li>Fade</li> <li>Fade In</li> <li>Fade Out</li> </ul>	Duration(S 0 • 1 1 .1 2 .2 3 .3 4 .4 5 .5 6 .6 7 .7 8 .8 9 .9 10 •	ec) Frames 1 2 3 4 5 6 7 8 9 11
	P	athname for tl	iis Object's Anir	mation Audio	Track
Inactiv	ie	•	-	Ma	iin Audio File
Animal	tion C	ycles Custor	n Zoom % Avoi	d Cursor Di 🔽	mpen Anim. E
Activate	O Pek O	bject Exit Timi iy (Sec) Din 10 1 No	aline Specificati ection Movem ne 🔀 None	ons ent Duration( 2 2 .0	Sec) Frames
Sp: Inactiv	ecifica e	itions for this	Object's Exit An	imation Audic	) Track at Audio File

FIG. 3D



FIG. 3E

319



FIG. 3F



## IPR2021-01146 Page 00168





FIG. 7









FIG. 9



FIG. 10



Ц Ц С





FIG. 12

18/18

1300 **Receive HTTP Header** 130 Query for **Device Characteristics Operator and Locale** 1301 NUMBER Generate JAD 210 **Response Director** Query and Receive URL for Matching Player 1303 **Device Appropriate Player Install Player Profile** Database Æ. 1305 **Player Build Process** Generate Players for all Abstraction Implementations

FIG. 13

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US2009/039695

International filing date: 06 April 2009 (06.04.2009)

Document type:	Certified copy c	of priority document
Document details:	Country/Office: Number: Filing date:	US 61/113,471 11 November 2008 (11.11.2008)

Date of receipt at the International Bureau: 14 April 2009 (14.04.2009)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse



## AND STREAM AND SYNAMAS (D MANA) AR (MA

#### TO ALL TO WILCON THESE PRESENTS SHALL COME:

#### UNITED STATES DEPARTMENT OF COMMERCE

**United States Patent and Trademark Office** 

April 13, 2009

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: 61/113,471 FILING DATE: November 11, 2008 RELATED PCT APPLICATION NUMBER: PCT/US09/39695

THE COUNTRY CODE AND NUMBER OF YOUR PRIORITY APPLICATION, TO BE USED FOR FILING ABROAD UNDER THE PARIS CONVENTION, IS US61/113,471



157238

Certified by

Under Secretary of Commerce for Intellectual Property and Director of the United States Patent and Trademark Office

PR2021-01146 Page 00177

PTO/SB/14 (07-07) Approved for use through 06/30/2010. OMB 0651-0032 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	XPR.002PR1	
		Application Number		
Title of Invention	SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES			
The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76. This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.				

## Secrecy Order 37 CFR 5.2

Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

## **Applicant Information:**

Applic	ant 1								Remove	
Applic	ant Authority 🖲	Inventor (	⊖Legal	gal Representative under 35 U.S.C. 117			7 OParty of Interest under 35 U.S.C. 118			C. 1 <b>1</b> 8
Prefix	Given Name			Middle Name		Fam	ily Name		Suffix	
	Steven			H.			Rem	ipell		
Resid	ence Informatio	n (Select Oi	ne) 💿	US Residency	y O M	Ion US Re	sidency		e US Military Service	!
City	Novato		St	ate/Province	e CA	CA Country of Residence i US			US	
Citizer	nship under 37 C	FR 1.41(b)	i U:	S	I	•				
Mailin	g Address of Ap	plicant:	1							
Addre	ss 1	38 Washin	gton Str	eet						
Addre	ss 2									
City	Novato	1			Sta	ate/Provir	nce	CA		
Postal	Code	94947			Country	i us		1		
Applic	Applicant 2									
Applicant Authority () Inventor () Legal Representative under 35 U.S.C. 117 () Party of Interest under 35 U.S.C. 11					C. 118					
Prefix	Prefix Given Name			Middle Name		Family Name		Suffix		
	David					Chro	Chrobak			
Residence Information (Select One)  US Residency  Non US Resider				sidency		e US Military Service				
City	Pleasant Hill		St	State/Province CA Country of Residence i US			US			
Citizer	nship under 37 C	FR 1.41(b)	i U:	s	I					
Mailin	g Address of Ap	plicant:	1							
Addre	ss 1	132 Shado	wood D	rive						
Addre	ss 2									
City	Pleasant Hill				Sta	ate/Provir	nce	CA		
Postal	Code	94523			Country	i us		1		
Applic	ant 3								Remove	
Applic	ant Authority •	Inventor (		Representativ	e under 35	U.S.C. 11	7	OParty of In	terest under 35 U.S.	C. 118
Prefix Given Name				Middle Name			Family Name			Suffix
	Ken						Brow	vn		
Resid	ence Informatio	n (Select Oi	ne) 💿	) US Residency	y () M	Ion US Re	sidency		e US Military Service	
City	San Martin		St	ate/Province	CA	Countr	y of R	esidence i	US	

#### PTO/SB/14 (07-07)

Approved for use through 06/30/2010. OMB 0651-0032 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Da	ta Shoot 37 CED 1 76	Attorney Docket Number	XPR.002PR1
Application Data Sheet St CFR 1.76		Application Number	
Title of Invention	SYSTEMS AND METHODS I	FOR PRESENTING INFORMAT	ION ON MOBILE DEVICES

Citizenship under 37 CFR 1.41(b) <sup>j</sup>		US				
Mailing Address of Applicant:						
Address 1 2485 Church		2485 Church A	Avenue			
Address 2						
City	San Martin	•		State	e/Province	CA
Postal Code 95046		95046	Co	untryi	US	
All Inventors Must Be Listed - Additional I generated within this form by selecting the Add			tional Inventor Inforn the Add button.	nation	blocks may b	e Add

## **Correspondence Information:**

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).					
An Address is being provided for the correspondence Information of this application.					
Customer Number	Customer Number 40280				
Email Address         svosen@PhDpatents.com         Add Email         Remove Em					

## **Application Information:**

Title of the Invention	SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES				
Attorney Docket Number	XPR.002PR1     Small Entity Status Claimed			ned 🗙	
Application Type	Provisional				
Subject Matter	Utility				
Suggested Class (if any)			Sub Class (if any)		
Suggested Technology Center (if any)					
Total Number of Drawing Sheets (if any)			Suggested Figure for Pu	ublication (if any)	

## **Publication Information:**

Request Early Publication (Fee required at time of Request 37 CFR 1.219)

Request Not to Publish. I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application has not and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

## **Representative Information:**

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Enter either Customer Number or complete the Representative Name section below. If both sections are completed the Customer Number will be used for the Representative Information during processing.

#### IPR2021-01146 Page 00179

PTO/SB/14 (07-07) Approved for use through 06/30/2010. OMB 0651-0032 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number. Attorney Docket Number XPR.002PR1

Application Da	ta Shoot 37 CED 1 76	Automey Docket Number	
Application Data Sheet of CI K 1.70		Application Number	
Title of Invention SYSTEMS AND METHODS		FOR PRESENTING INFORMAT	ION ON MOBILE DEVICES
Customer Number	40280		

## **Domestic Benefit/National Stage Information:**

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78(a)(2) or CFR 1.78(a)(4), and need not otherwise be made part of the specification.					
Prior Application Status Remove					
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the Add button.					

## **Foreign Priority Information:**

This section allows for the applicant to claim benefit of foreign priority and to identify any prior foreign application for which priority is not claimed. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(a).					
Remove					
Application Number	Country <sup>i</sup>	Parent Filing Date (YYYY-MM-DD)	Priority Claimed		
			🔾 Yes 💿 No		
Additional Foreign Priority Data may be generated within this form by selecting the Add button.					

## **Assignee Information:**

Providing this information in the application data sheet does not substitute for compliance with any requirement of part 3 of Title 37	
of the CFR to have an assignment recorded in the Office.	

Assignee 1	Remove						
If the Assignee is an Organization check here.							
Prefix	Given Name	Middle Name		Family Name	Suffix		
Mailing Address Information:							
Address 1							
Address 2							
City			State/Province				
Country <sup>i</sup>	Postal Code						
Phone Number			Fax Nun	nber			
Email Address				Ŀ			
Additional Assignee Data may be generated within this form by selecting the Add Add							

## Signature:

A signature of the applicant or representative is required in accordance with 37 CFR 1.33 and 10.18. Please see 37 CFR 1.4(d) for the form of the signature. EFS Web 2.2.1 IPR2021-01146 Page 00180
#### PTO/SB/14 (07-07) Approved for use through 06/30/2010. OMB 0651-0032 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	XPR.002PR1
		Application Number	
Title of Invention	SYSTEMS AND METHODS F	FOR PRESENTING INFORMAT	ION ON MOBILE DEVICES

Signature	/Steven R. Vosen/		Date (YYYY-MM-DD)	2008-11-11	
First Name	Steven	Last Name	Vosen	Registration Number	45186

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450**.

## **Privacy Act Statement**

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

- The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
- 2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
- 3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
- 4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
- 5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
- 6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
- 7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
- 8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
- 9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Acknowledgement Receipt			
EFS ID:	4267923		
Application Number:	61113471		
International Application Number:			
Confirmation Number:	9197		
Title of Invention:	SYSTEMS AND METHODS FOR PRESENTING INFORMATION ON MOBILE DEVICES		
First Named Inventor/Applicant Name:	Steven H. Rempell		
Customer Number:	40280		
Filer:	Steven R Vosen		
Filer Authorized By:			
Attorney Docket Number:	XPR.002PR1		
Receipt Date:	11-NOV-2008		
Filing Date:			
Time Stamp:	17:59:24		
Application Type:	Provisional		

# Payment information:

Submitted with Payment yes		yes				
Payment Type	Payment Type		Credit Card			
Payment was successfully received in RAM		\$515	\$515			
RAM confirmation Number		11332				
Deposit Account						
Authorized User						
File Listing	:					
Document Number	Document Description	File Name	File Size(Bytes)/MultiPagesMessage DigestPart /.zip(if appl.)			

1	Provisional Cover Sheet (SB16)	rovisional Cover Sheet (SB16) XPR002PR1ProvisionalCoverfsh eet.pdf 4dd9c32dd02a9cd33Bb000caa293bfe5ad 0b21		no	3	
Warnings:	Warnings:					
Information:						
			964561		5	
2	Application Data Sheet	XPR002PR1ApplDataSheet.pdf	c98a6bff25b40c6ee8c721f241c8f714717b3 0d0	no		
Warnings:						
Information:						
2			8400894		225	
5	specification	Xmo_0G_05e.pdi	dfecbfa955464ae5e48e72b2b12dad970c6 9583d	no	525	
Warnings:						
Information:						
4	Foo Mortschoot (PTO OS)	fac info ndf	31346		2	
4	ree worksneet (PTO-06)	iee-inio.pai	f7ec785d1f617e59090a2f3a78e49d287947 2af9	no	2	
Warnings:						
Information:			1			
		Total Files Size (in bytes)	10.	396355		
This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.   New Applications Under 35 U.S.C. 111   If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.   National Stage of an International Application under 35 U.S.C. 371   If a timely submission to enter the national stage of an international application is compliant with the conditions of 35   U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.   New International Application Filed with the USPTO as a Receiving Office   If a new international application is being filed and the international application includes the necessary components for an international application filed with the USPTO as a Receiving Office   If a new international Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application includes the international Application scorering national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.						



 $\ensuremath{\textcircled{}}$  2008 Express Mobile, Inc. All Rights Reserved.

## User Guide TOC

# **Getting Started**

Installing the Publisher	01.04
Log In	01.05

# Chapter 1: Overview of the Interface

A:	Menu Bar	01.05
B:	Toolbar	01.05
C:	Canvas	01.12
D:	Layer Inspector	01.12
E:	Resource Inspector	01.16
F:	Page Viewer	01.17
G:	Palette Docking	01.21

# Chapter 2: Working with Pages

A:	Master Page	02.04
B:	UI Implementation	02.06
C:	Application Pages	02.08
D:	Alerts	02.14
E:	Libraries	02.16
F:	Saving Files	02.17
G:	Enhanced Navigation Specifcation	02.18

# Chapter 3: Introduction to Objects

A:	How to Create	03.05
B:	Keyboard Operations with UI Objects	03.05
C:	Persistent Variables	03.06
D:	Layout by Pixel vs Layout by Font	03.17
E:	Suppression of Selection Rectangle	03.19
F:	Using Object Libraries	03.20

# Chapter 4: Working with Navigation Objects

A:	Creating a Launch Strip	04.04
B:	Working with Launch Strips	04.04
C:	Working with Complex Lists	04.07
D:	Inter-Application Navigation	04.08
E:	Bubble Help Color	04.08

# Chapter 5: Working with Message Objects

A:	Working with Chat	5.04
B:	Working with Phone Calls	5.07
C:	Working with SMS Messages	5.09
D:	Working with Rich Media Alerts	5.11

# Chapter 6: Working with Business Forms

A:	Introduction	06.04
B:	Text Fields	06.05
C:	Always On Text Fields	06.06
D:	Lists and Dropdowns	06.09
E:	Text Entry	06.23

# Chapter 7: Working with Media Objects

A:	Pictures (Bitmap Images)	07.04
B:	How to Edit Using Photo Enhance	07.05
C:	Slide Shows	07.09
D:	Video	07.16
E:	LBS Maps	07.19

# Chapter 8: Working with Vector Shapes

A:	How to Create	08.04
B:	How to Create a Group	08.06
C:	How to Edit	08.07

# Chapter 9: Working with Text Objects

xpressmo.com

IPR2021-01146 Page 00188

A:	How to Create	09.04
B:	How to Edit	09.05

# Chapter 10: Working with Child Objects

A:	How to Create	10.04
B:	How to Select	10.06
C:	Behaviors	10.07

# Chapter 11: Working with Web Components

A:	Supported Objects	11.05
B:	How it Works	11.07
C:	Behaviors and UI	11.10
D:	Adding a Web Component	11.11
E:	Completing a Web Component with the "Generate UI" Option	11.13
F:	Completing a Web Component with the Existing UI Objects	11.19
G:	Persistent Variables	11.23
H:	Enhancing Web Components with Auto-Generated Persistent Variables	11.25
l:	Restrictions for Editing UI Objects	-
	related to Web Components	11.29
J:	Working with Control Lists	11.31
K:	Working with Collection Lists	11.38
	xpress	mo.com

L:	Collection List Keys and	
	Complex List Indexes	11.40
M:	Sharing a Web Component	11.42
N:	How to create a Web Component Model	11.47

# Chapter 12: Working with Object Styles

A:	Introduction	12.04
B:	Object Styles	12.05

# Chapter 13: Animations & Timelines

A:	Introduction	1 <b>3.04</b>
B:	How to Define	13.06
C:	Child Object Animations & Behaviors	13.10
D:	Defining Object Entry Timelines	13.11
E:	Defining Object Exit Timelines	13.12
F:	Tips	13.13

# Chapter 14: Working with Events

xpressmo.com

## IPR2021-01146 Page 00190

A:	Child Object Events	14.04
B:	Exit Events	14.05
C:	Select and Fire Events	14.10
D:	Waypoints	14.13

## Chapter 15: Testing Your Application

Testing an Application created in Publisher 15.04

## Appendix A: Mapping for Direct Text Entry

Appendix B: Object Animations

Appendix C: Web Component Models

Appendix D: Landable Attributes

xpressmo.com

IPR2021-01146 Page 00191

Publisher Interface

User Guide TOC

<section-header><section-header><section-header><section-header><section-header><section-header><section-header>

Installing Publisher

© 2008 Express Mobile, Inc. All Rights Reserved.

# Getting Started

Installing the Publisher	01.04
Log In	01.05

# Getting Started : Glossary

JRE

### Java Runtime Environment (JRE)

Quicktime

Apple's popular and comprehensive Vide and Audio Player. Compatible with Java.

Getting Started.4

## Installing Publisher

Publisher Client can be installed on Windows XP SP2 or higher, and on Mac OS X 10.1 or higher. Please consult your System Administrator about the compatibility of your machine's OS. If you wish to run Publisher on a machine that does not already have Publisher installed, obtain a copy of the Publisher Client installer from your System Administrator. Also, confirm that the account you are using on your machine has the privileges required to install a new application.

Make sure that the following support software is installed and is the correct version for the Publisher Client which you plan to install:

- Java Runtime Environment (JRE)
- QuickTime
- Operating System

For details on installation requirements and procedures, please refer to the document "Publisher 2.1 Client Install Guide".

## Log In

Once Publisher Client is installed, open Publisher by finding the desktop icon named "Publisher Client" or by selecting from the taskbar Start  $\rightarrow$  All Programs  $\rightarrow$  Publisher Client  $\rightarrow$  Publisher 2.1. You should see a log in window as shown here.

Obtain the following information from your System Administrator prior to first log in:

- Username
- Password
- · Server to which the client connects

Select "Submit" once you have entered the required information. In a few seconds, you should see the Publisher canvas page.

On subsequent log ins, Publisher client will remember the last used username and server, so all you will need to type in is your password.

xhig221110	
AUTHORING PLATFORM	2 2
kername:	**
ssword:	
server: Submit Cancel Help	
	AUTHORING PLATFORM Dername: Password: Server: Submit   Cancel   Help

١

xpressmo.com

## IPR2021-01146 Page 00198

# **EXPRESS MOBILE** Üser's Guide

Chapter J

Interface Overview

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 1: Overview of the Interface

A:	Menu Bar	01.04
B:	Toolbar	01.04
C:	Canvas	01.11
D:	Layer Inspector	01.11
E:	Resource Inspector	01.15
F:	Page Viewer	01.16
G:	Palette Docking	01.20

# Chapter 1: Glossary

Application Page	A page that is available to be visited through any navigation event. Application Pages inherit all the attributes of the Master Page, unless that attribute is specifically changed during an editing session.
Master Page	The initial definition for all application pages. Any additions or changes to the master page will be immediately reflected in all application pages, unless prior editing of that application page modified that par- ticular attribute.
UI Parent object	Any object that is not a child object. Parent Objects can be selected when running the application.
Ul Child Object	Any object that was created by first selecting a Parent UI Object and then creating a Child Object. Child objects are always part of the same drawing layer as its parent object, but will draw first. Child objects themselves are not directly selectable when running the application.
Author's Private Page Library	The author can at any time save any application page into their Private Page Library. This library is restricted to the author that created this library, and is available during any subsequent editing session and these pages can be imported into any application.
Xpressmo Public Page Library	This library contains pages that are available to all authors of the orga- nization that is licensing the Express Mobile Publisher Platform.
Page List	The Page List is part of the Layer Inspector. It displays all the pages currently available in the application.
Edit Page Dialog Box	This dialog box offers all the choices that is available through the Re- source Inspector of a Page, but these changes are not applied to the page until the Edit Page Dialog Box operation is completed.
Style Object	Style objects contain the initial default values that will be applied to any UI Parent or UI Child Object when it is first created. Any changes to a style object will be reflected in all the UI Objects that are attached to this style object, unless, during an editing session, that particular attribute had been changed in the UI Object.

xpressmo.com

## IPR2021-01146 Page 00201

Dynamic binding	This is the technique that permits the author to connect many attri- butes of UI Objects to a particular data base field on a Server. When running the application, the current value in the referenced data base will be immediately applied. During the application session, any other real time changes to these values in the referenced data base will again be immeditely displayed.
Resource Inspector	The primary Interface Panel for interactively working with UI objects that have been placed on the Canvas.
Page Viewer	One one several floating palettes that provide direct feedback and interactions with a set of objects and functions. The Page Viewer will, in various ways, draw one, some, or all the current application pages at the same time as the current canvas is available.
Hot Zone	This is the space in the current defined view port (Page Size), that, if Page Scroll Bars were drawn, they would lie in this hot zone. This helps the author to place objects so that the need for Page Scroll Bars is minimized.



A. Menu Bar

The Menu Bar offers a comprehensive way to work with the Publisher.

## B. Toolbar

The Choices under the Toolbar, including the floating palettes, represent a convenient way of accomplishing the same functions as the Menu Bar.

#### 1. Add pages to the Canvas

With your mouse Select the Insert Page Icon to bring up the "Insert Page" dialog box. If Append is not checked then the page will be inserted in front of the current page that is in the Canvas. The Page name must be unique for the current application. If not, then the name will turn red until it becomes unique again.

ſ	· · · · · · · · · · · · · · · · · · ·
	Append:
	· [
	OK Cancel Heid

xpressmo.com

Chapter 01.5

## IPR2021-01146 Page 00203

#### 2. Add UI Objects to any page in the canvas

The Toolbar exposes various categories of UI Objects, all of which can be selectively placed on the Canvas.

a) Navigation

Select the Navigation Icon to access Navigation Objects that become a starting point for designing a navigation model for your application. In addition to changing the many attributes for any of these Navigation Objects to create very different results, the Events Model (See Chapter 14) can also be used. Also see (Working with Navigation Objects in Chapter 4);



b) Messages

Select the Messages lcon to view the "Message Objects" menu. There is a very large set of message Objects that can be used for communication though content such as:

- 1) Chat
- 2) IM
- 3) Placing Phone Calls from Lists that could have been populated from the phone address book or some internet-based Contact Management System.
- 4) Sending SMS messages to one recipient, or a group of recipients, again from lists that been populated from the phone address book or some internet-based Contact Management System.
- Sending Rich Media Alert Messages that could include any page and/or widget that is available to that consumer.



Chapter 01.7

#### c) Forms

Select the Forms Icon to select a Forms Object. The Publishing tool offers a very wide set of intelligent and highly extensible business forms. "Working with business forms" (Chapter6) and "Working with complex Lists" (In Chapter 4) should both be consulted as Business Forms can also become powerful Navigation Objects. Also "Persistent Variables" in Chapter 3 describes how Business Forms can act as a simple front end to a Mobile Device resident database.



d) Media

Select the Media Icon to select the Media Objects menu.

See "Working with Media Objects" in Chapter 7.



#### e) Shapes

Select the Shapes Icon to access the Shapes Object menu. There are 4 drawing tools, various groups of standard vector objects, as well as a library for full 2D vector graphic images.



Fig 1-10

#### f) Text

Select the Text Icon to show the Text Objects menu.

- 1) Text Buttons (single line text objects that scale automatically to the length of the text string)
- 2) Paragraphs (multiple line, and multiple paragraph) text objects that work similar to that of a Word Processor, and will grow automatically to fit the enclosed Text.
- 3) Search Response (a shortcut to the default Search Web component.
- 4) RSS Display (a shortcut to the RSS Display Web Component).
- 5) Display Time and/or Date Variables in various formats.



Fig 1-11

#### g) Palettes.

Select the Palettes Icon to see the Palettes Objects menu.

Any of up to 5 floating palettes can be placed on the display, in any desired location. Once placed, there will be memory so that if you move the Publisher Window, or if you exit and then restart the Publisher, the Floating Palettes will remember their location relative to the Publisher's Window.

The Page Viewer is discussed later in this chapter. Section  $\ensuremath{\mathsf{A.1.a}}$ 



#### 1. The View Palette:

The View Palette permits the following functions to be performed.

a: Edit or Preview:

The default is Edit. Selecting the Play Icon will close The various Panels and Inspectors and immediately begin to preview the application inside the Publishing tool. (In this mode certain functions, such as Exit Events, are not available)

11	$\triangleright$	Q	Q	Loom Grid	Scow Hot Lone	<b></b>
Stop:	Record •	10.0		No Gent *	Zoom Page Viewer	1.0994

Selecting the Stop Icon will return the Publisher to the Edit State.

The author can also selectively preview the current page or preview the entire application by selecting either "Go: Current Page" or "Go: All Pages" respectively. If the author selects "Go: Launch" a browser window will open and the application will play in a manner similar to how it would work on a phone, except that the mouse is available as a selection tool. If the author wishes to simultaneously create a Desktop version of the Application, then Launch will behave identically as it will in these environments. The default browser that you are using will be selected but be sure that the Java J2SE 6.0 or higher plug in is available.



#### b: Zooming:

The default zoom level is 100%. Touching the + or - lcons will increase or decrease the zoom in increments of 10%. Holding down the mouse while on these icons will cause a continuous zoom until the mouse is released. Alternatively, the author can pick any zoom level form 20% to 800% by selecting the desired zoom level from the dropdown menu.

c: The Snap-to Grid

If one of the 3 grid settings is selected from the Zoom Grid Drop Down, then all objects will snap its upper left hand corner of its bounding box to the nearest grid point. The graphic to the right shows Medium (40 pixel) grid setting. Additional Grid controls are available by selecting "Grid..." under "View" in the Menu Bar.

d: Show Hot Zone

See Chapter 2 (Working With Application Pages).

e: Zoom Page Viewer

See "The Enhanced View Palette" discussion, in the Page Viewer Section later in this chapter.

	1	I			
				8	
				×	
 				·····	
l	A text	button	Į	8	
				1	
				Ĭ	
				Fig 1-1	15

#### 2. The Command Palette:

This palette is a convenient shortcut for the most common commands that are also available under "File" and "Edit" from the Menu Bar. The "Group" and its counterpart "Ungroup" icons become available as soon as more than one object is selected at the same time. This is accomplished by holding down the "Shift" key and then selecting the desired objects that you wish to group.

D	ø	6	Æ	°0	Ô	×	ĸ		Select All	Close
---	---	---	---	----	---	---	---	--	------------	-------

٠

3. The Color Palette:

The Color Palette, in its default state, shows the currently selected color that can be applied, with a single touch, to up to four color attributes for the currently selected object.

- Fill
- Line
- Text
- Frame

Selecting any of the four arrows will expand the color palette as shown on the right. These additional controls permit the change of the Color for that attribute. These choices will be remembered between editing sessions.





4. The Shape Palette:

The Shape Palette offers another convenient way for placing shapes on the Canvas (See Shapes in Section B.2.e)

Draw Tools	Basic	Arre	2 and	Lines	Library	
R	<1	N	m			

## C: Canvas

The canvas is where all the work is performed for building or modifying an application. You can work with any object on the canvas directly with the mouse, and indirectly through the Inspector Panel or the Page/Objects Panel.

The Canvas can be set to any zoom level form 20% to up to 800% of the actual size. If the page becomes larger than the visible canvas then the scroll bars become enabled.

## D: Layer Inspector

The Layer Inspector controls:

- 1. The Page (Master Page, Application Page or Alert Page) you want to work with.
- 2. Permits various operations for Parent and Child Objects that are assigned to that Page:
- 3. Permits various operations for Object Styles:
- 4. Navigating between Page types and Object Styles
  - a. Toggling the Pages Icon:

This will switch between the Master Page and Application Pages It will also deselect Object Styles and Alerts

b. Touching the Object Styles Icon:

This will open the Edit Styles Dialog Box and deselect any Master, Application or Alert Page.

c. Touching the Alert Icon

This will select an Alert Page and deselect any Master Page or Application Page.

Pages	
3 Home_A	
3 E /	
Objects	
20: background	
19: Traffic_image	
18: weather image	
17: WeatherDetail	
P16: Iraffic_textarea	
15: Image Object_19	<u> </u>
14; gaewsu1 13: gaow602	
13. ynewsuz 12: anguens	
11: mewe04	
10: Image Object 17	+
	n Ga
Object Styles	
Default	, <b></b>
Header	Ē
Title	
Body	
Item	
Credit	
wewer Mardunes	
a 5 1	6
Alerts	6 c
messagesent	
-	
B A A 2	

Chapter 01.13

- 5. In addition, touching any object or icon in the related list for:
  - a. Pages/Object Panel
  - b. Object Styles Panel
  - c. Alerts Panel

will automatically select that panel.

a: Pages/Objects Panel

Page Operations:

- 1. All Selection activities for:
  - a. Any Application Page
  - b. The Master Page
  - c. Any UI Parent object.
  - d. Any UI Child Object
- 2. Common Page Operations



- a. Import from either the Author's Private Page Library or from the Xpressmo Public Page Library.
- b. Insert a new page, inheriting all the attributes of the Master Page, and place this new page at any location in the Page List.
- c. Edit the currently selected page, by working with the Edit Page Dialog Box. All the functions of the Resource Inspector are available, but are not applied to the actual page until exiting the Edit Page Dialog Box.
- d. Delete the selected Page.
- 3. Common UI Object Operations
  - a. Change the drawing order layer to:
    - i. Bring to the front
    - ii. Send to the back
    - iii. Bring to the front one layer
    - iv. Send to the back one layer.
  - b. Edit the currently selected page, by working with the Edit Page Dialog Box. All the functions of the Resource Inspector are available, but are not applied to the actual page until completing the entire editing process.
  - c. Hide (and then reshow) the Selected Object. This facilitates working with objects whose display is, at least in part, obstructed by another UI Object.
  - d. Delete the Selected UI Page Object

b: Object Styles Panel

Similar to working with Pages, Object style Operations include:

- 1. Import from either the Author's Private Object Style Library or from the Xpressmo Public Object Style Library
- 2. Insert a new Object Style.

The styles can be inherited form the currently selected object (as shown to the right with the "gnews01" UI Object selected, or from a previously defined Style Object.)

9: frattic image 0: www.char.image	1	d a Neve style	- <b>1</b>	Contraction of the second
/: Weather Dalad 16: Isalite_lextanoa	- Soya	e tiame	Basad Do	
5: Image Object 19			2000W 101	·
A gnows07			Sales Million	
2: spews03	1200 88	and a second	Defash	- 63
1: gnews04 0: image Object 117	. 1942		Tick	
			Bich	- 1
a a a a 🖉 🖉	U .	9	Didi	
Object Styles		e, zoo	1. sodit Nes is sta	
in a state of the	<b>7</b> .7			

3. Edit the currently selected Object Style by working with the Edit Style Dialog Box.

Style attributes can be assigned many attributes, including the look, and behavior of any object that inherits these objects. In addition, List Layout Styles can be created or changed (See the Section on Complex Lists)



Fig 1-23

c: Alerts Panel

Alert Pages can have many of the attributes of Application Pages, but they are only activated through an Event (See Working with Events in Chapter 14) and will be superimposed onto whatever the current Application is. (See the section "rich media Alert Messages" in Chapter 5).

- 1. Alert Pages all have transparent backgrounds, and they function as an template overlay.
- 2. Alert Pages can have dynamic binding to real time content.



## E: Resource Inspector

The Resource Inspector, whether working with pages or UI Page Objects, has five Standard Tab Selections.

1. The Settings Tab:

When activated, the settings dialog permits basic configuration of the selected object. These include:

- a. Name
- b. Size
- c. Location
- d. Navigation and Visual Settings
- e. Other

Depending upon the type of object, numerous other attributes could be exposed.

2. The Events Tab:

This includes all end user interactions and time based operations. (See Chapter 14)

3. The Animation Tab:

This includes all animations and timelines. (See Chapter 13)

4. The Color Tab:

- Settings Events Animation Color Dindings Web Page Restore Octaolis Enab Idelault f Navigate by Object Page Icon Settings 12 Add Page Icon Unavel ett -Page Size Settings Reminero Paras laces es VV Soft Key Commanda \$71.797.5.986 1.4.8 Master Page Objects Doline Scrott Har 176x186 \$78,220 if Phone: 176x144 A Standard: 420x600 A Lendscape: 460x5-Std Landscape: BOOx600 Std Portrait: 6001800 Size: 2000x800x Which 240 ¥ 500 [438 Heigh 260 134 taac live Background A Object Selected: Page Fig 1-25
- This includes all the possible color attributes. These vary significantly by object type.
- 5. The Bindings Tab:

This is where all Web Component Operations are defined and all dynamic binding settings. (See Chapter 11)

Chapter 01.17

## F: Page Viewer

#### 1: Activation:

Methods to select the page viewer pallete:



vas. Screen shot to the right shows the choice from the "View"

menu.

Screen shot to the left shows

the choice if there is a right

click when on the working can-



Finally the Page Viewer can be activate the Palettes menu.

- In addition, if the Page Viewer had been activate when the last Publishing Session was terminated, then the Page Viewer will automatically appear, with the following memory:
- The Position and Size of the Page Viewer Dialog Box.
- The Zoom Level that had last been selected.

#### 2: Dismissal:

There are four ways to dismiss the Page Viewer:

- a. From the Palettes Menu.
- b. From the View Menu.
- c. By Touching the Close Icon.
- d. By Closing the Page Viewer Dialog Box.



#### 3: Positioning and Zooming:

The Page Viewer will act like a floating palette, much like the other palettes. However, the user can freely resize the Page Viewer. Immediately the Page Viewer will reflow the page thumbnails to the changed container size. If the height of the Page Viewer is less than 150% of the height of a page thumbnail, then a Horizontal Scroll Bar will be displayed (if necessary) and all the thumbnails will line up in a single row. See the example below which has set the zoom level of the page thumbnails set to 30% of the original, and has resized the Page Viewer container into a horizontal strip.

C Daro Monarda I: Guido A Daro Monarda C Daro Monarda I: Ilano Tata I: I	stativeny 4: Teplient 7 5: Mail	6: Writter - 2: Al	
11 File Cal Vew ProjeC Occor Panel Dupor Rays Navigation C C C Cal Cal Cal Cal Cal Cal Cal Cal Ca	e Pique State Hat State Hat Manuages Farma Merila 32	La T.	Autours
Dijerts Objerts 7: skop intro 1: astikile Verbcalk on Stip	User's Guide		Vitto Page Hame Enable Li Júante C Navigale by Object SciGante - Page Skie Settings Droman Vlandow GXCA Proces 200/250
oberasje oberasje		intro	CVG Phone: 176:103 CVG Phone: 176:103 CCF Phone: 176:103 CCF Phone: 176:103 CCF Phone: 176:20 CCF Phon

The zoom level can be changed as required from inside the Page Viewer.

Fig 1-29

If the height of the Page Viewer is greater than 150% of the height of a page thumbnail then the thumbnails will display as a grid, with a Vertical Scrollbar available.

() property (12) storing the or	Constitution of the second	Acres 3. Pape Verner		2996 Q	
	<u>_</u> .e. <u></u> .				
1 (2007 P-0:// \$10 Depen	Usar's Guide				
2 skojatra 1 salate Ventuseconsela	Operating the AOL Moose Ports				
ata Jan 2019 - 10			<b>4</b> 2007 11 28 -		
Clipert Uybr		CERTIT			
Default Heater Bate Dady Uran Certor Certor Certor					
Gardon Burt	00084%				

Fig 1-30
The Author can change the zoom level to 200% or even up to 300% so that the Page Viewer acts as a magnifying lens.



4: Preferred Size:

When opening Publisher, if the Page Viewer is visible, it will resize itself to a preferred size. Specifically, the size will adapt itself to the appropriate size to display the default page. When additional pages are added to the application, the Page Viewer will resize itself to the the latest page dimension.

#### 5 : View Palette:

The View Palette now has 2 new entries:

- 1. Show Hot Zone
- 2. Zoom Page Viewer

Changing the Show Hot Zone setting or the Page Viewer Zoom Level will be immediately reflected in the Page Viewer Dialog Box.





6: Editing with the Page Viewer activated:

All visible editing operations will immediately become visible in the Page Viewer.

7: Using the Page Viewer for Navigation:

The page number and page name will usually draw over the upper left hand corner of the Page thumbnail. However, when the mouse is over a particular page thumbnail, the page number and name will attach itself to the cursor.

Selecting a page image within Page Viewer will switch the canvas to that page.

## G: Palette Docking

#### 1: Palette Docking

There are currently 5 palettes available in the Pub Tool.

- Page Viewer
- Command Palette
- View Palette
- Color Palette
- Vector Palette

All of these palettes can be dragged to any position on the screen. They can selectively set to visible or hidden.

The Page Viewer Palette can also be resized.

When dragging or resizing the Publisher windows, all visible palettes will move in synchronization with the Pub Palette Frame (they are effectively docked to a fixed position in or relative to the Publisher Palette Frame). Any dragging and/or resizing of palettes will have no effect on the position or size of the Pub Tool Frame.

As with the Publisher frame itself, the location and size of all the visible palettes will be remembered when the Author exits from the Publisher. These positions and sizes will be restored when the Author returns to the Publisher.

xpressmo.com

## IPR2021-01146 Page 00220



© 2008 Express Mobile, Inc. All Rights Reserved.

3

# Chapter 2: Working with Pages

A:	Master Page	02.04
B:	UI Implementation	02.06
C:	Application Pages	02.08
D:	Alerts	02.14
E:	Libraries	02.16
F:	Saving Files	02.17
G:	Enhanced Navigation Specifcation	02.18

Chapter 02.3

# Chapter 2: Glossary

Launch Strip Attribute	The settings for each Icon in the Launch Strip. Usually they are links to differnt pages or applications.
Uber-Application	The first application that is executed that activated by the phone. Usually this is defined by the author.
Sub-Application	An application is is called either from the Uber-Application or from another Sub-Application.
Alert	A type of page that is superimposed over the current application page.
Soft Key Command	These are commands that become available through touching a Soft Key button on the mobile device.
getParentObjectSelected	An example of an Interface Method that is called, in this case, when an Object is selected. These Interface Methods are very useful for working with Web Components, Express Mobile APIs, and for collect- ing analytics.
PDL Image Table	A table of all the Application and Alert Pages that have currently been downloaded and are available.
dimPage	A screen that is drawn over the page when the page is temporarily inactive. This often occurs when a launch strip is active or an Alert page is active.

### A: Master Page

1. Defines Default Settings (at the page level) for all new pages.

This means that all the attributes that can be set in the "Edit Page" dialog box and/or the Page Resource Inspector will have their initial default settings defined by the Master Page.

2. Repeating Objects.

Any objects defined on the Master Page will appear on every application page. The drawing order for an application page will be:

- a. The first objects drawn will be from the Master Page, and drawn in the order defined in the master page.
- b. All the application page specific objects will then be drawn in their drawing order.
- c. Finally, if defined in the Master Page, the Launch Strip(s) will be drawn.

#### 3. Editing Operations:

a. Editing the Master Page:

To work with the Master Page select the "Master Page" entry in the Page List or by Selecting the "Master Page" command under the file menu. The currently viewed page will now be the "Master Page. All operations that are available to an application page will be available to the Master Page. Any additions, deletions, or changes to objects on the Master Page will immediately take effect for all application pages in the application.

b. Editing Master Page Objects on an Application Page.

Only Launch Strips can be selected and edited. All the other objects from the Master Page will be drawn but not be selectable.

#### c. Applying a Current Page as the Master Page

Select "Set Master Page" under the Pages menu. The current page's objects will be appended to the Master Page in their current drawing order. Any page attributes that the current page has set to override the Master Page's attributes will be transferred to the Master Page.

d. Selectively Removing Master Page Objects form an Application Page.

When the "Master Page Objects" button is selected from the Resource Inspector the Visibility of Master Page Objects Dialog Box will appear. All existing Master Page Objects will appear in the List on the right, and any Suppressed Objects will appear in the List on the left. The Suppressed Objects can be restored by selecting the Object in the list and Selecting the "Restore Object" button.

Fig 2-1 shows what happens when a Master Page Object was selected and the "Remove Object" button was pressed.

MasterPage Image Object 2 MasterPage Button Object 2 MasterPage Button Object 2 Restore Object Remove Object	Suppressed Objects	Master Page Objects		
Restore Object	MasterPage Image Object 2	MasterPage Image Object_2		
Restore Object				
	Restore Object	Remove Object		

- e. Current Master Page Restrictions.

Soft Key Commands are disabled for Master Pages.

4. Effect on Launch Strips.

Launch Strips are part of the Master Page. The only difference between Launch Strips and other repeating objects are that the Uber-application's Launch Strips will be inherited as repeating objects for all sub-applications. Any Launch Strips defined in a sub-application will inherit all the icons from the uber-application, with its own icons appended. These sub-application specific icons will only be available while the sub-application is the current application for the player. While a sub-application is running, its Launch Strip attributes will take precedence over the uber-applications Launch Strip attributes, if both sub-application and uber-application define the same-named launch strip attribute.

## B: UI Implementation

The Master Page can be selected in the [Master Page] entry in the Page List or by Selecting the "Master Page" command under the file menu. The result will be the graphic below.

To return to an Application Page or an Alert Page either Select the "Master Page" or "Alerts" icons or by Selecting a page or alert from their respective dropdown menus.

1       default         Image: Construction of the second seco	Master Page   Image: Image in the image i
Object Styles     Default     Header     Title   Body   Item   Credit     Oredit     Alerts   SystemAlert   Image: SystemAlert     Image: SystemAlert	<ul> <li>Fig 2-3</li> <li>Fig 2-3</li> <li>These icons allow the Author to change the drawing layer, import, create, delete and edit Objects on a Page. An alternative method for editing an Icon Style is simply fire on the "Object Styles" Icon and the Edit Dialog Box for the currently selected style will appear.</li> <li>The up and down arrow icons permit the navigation through application pages and alert pages respectively.</li> </ul>

Chapter 02.7

Fig 24

Selectively hiding an Application Page Object works as follows:

1. Hiding Objects

Select the object and then Select the "Eye" lcon. A non-alpha character similar to a lozenge character will appear to the left of the object name. When a hidden object is selected the "Eye" icon appears in a grayed out state. (See Fig 2-4)

2. Restoring Hidden Objects

Select the grayed out "Eye" icon as shown in Fig 2-4.

3. Effect of Hiding Objects

Objects	
13: Image Object_4	6
= 12: WeatherDetail	0
#11: icnSunny 👔 🕈	
10: Image Object_5	
9: tmage Object_6	
8: Image Object_7	$\mathbf{\overline{\cdot}}$
	<u>ک</u>
and we shall a state	5. <b>(A. ).</b> (BAK, K. )

A hidden object will only have effect during the current editing session. To hide a Master Object for a page during execution refer to "A.3.d" above.

S CHARTER STORES

# C: Application Pages

Edit Insert Page Goto Page Copy Page Page Viewer	Pages         1       detault         2       2         3       2         6       2         7       2         8       2         9       2         9       2         1       1         1	FIL:       Edit       View       Project         New       Ctrl+N       Open       ▶         Close       Quit       ■
Navigation • Messages •	1. The Popup Menu to the left which appears by using the right mouse button when	Save Save As Ctrl+S
Shapes Shapes Library Photo	over the Canvas. 2. The Page Control from the "Pages/Objects Panel"	Import Page Import Alert Import Widget
Video	above. 3 The Page oriented com-	Import List
Slide Show SlideShow Library	mands from the File Menu to the right.	Master Page Copy Application
Virtual Tour Virtual Library	4. The Pages menu below.	. Fig 2-7
Mobile GIS Service	Pages Styles Help	
Button Paragraph	Insert Page Ctri+I Delete Current Page	
Variables •	- Goto Page Define	
Forms +	Copy Page	
Select All	Set Master Page	
Deselect All	rg 2-8	

1: Page Resizing

Any page can be set to a custom size. If not, then the size of the master page will be inherited. Automatic generation of QVGA and QCIF applications is supported.

2: UI Implementation

File Edit Panel	View Project + Insert Page	Objects Navigatic	Events Pages	Styles Help	Med
1  default ⑥ 준 Ø Objects	Pages				

Fig 2-9

All pages have a hot zone [or a safe area] that is represented by two gray dotted lines. As long as the Author does not have a selectable object intersect these boundaries then a maximum of one scroll bar will be drawn assuming that one of the pages dimensions are equal to, or less than that of the device's canvas.



Page Size Settings represent the size for the currently selected page. This size is initially inherited from the Master Page, but can be changed by editing the width and/or height fields. If the resulting size does not match a standard form factor then "Custom Size" will be selected.

An Icon can be added so that if the Page is saved as a Widget then this Icon can become part of a Widget Selection list.

Objects that were defined in the Master Page are automatically placed in each application page. However, touching the "Master Page Objects" button will open up a dialog box where Master Page inherited objects can be selectively suppressed for a given application page.

An Audio File can be attached to the Page, so that when the Page is active that Audio will play or loop, depending upon the setting. Persistent variables can be added to a page. They are used so that the background audio could be defined dynamically trough a Web Component.

In addition, there is a button "Define Scroll Bar" which gives the author the ability to custom define the scroll bars for the currently selected page, if any. These values are also initially inherited from the Master Page.



Fig 2-11

The above Dialog Box gives the author the following choices.

- Horizontal Scroll Bar width (from 6 to 20 pixels)
- Vertical Scroll Bar width (from 6 to 20 pixels)
- Scroll Bar Background Color
- Elevator Background Color
- Scroll Bar Frame 1 Color
- Scroll Bar Frame 2 Color
- Elevator Frame 1 Color
- Elevator Frame 2 Color
- Draw Full Page
- Suppress Scroll Bars

The six colors permit a scroll bar definition that mimics a Windows or Java J2SE scroll bar. Any of these color attributes can be changed and the feedback will immediately reflect that color.

The Scroll Bar widths will change the width of the respective hot zone, and will establish the width of the scroll bar for that particular page.

If Draw Full Page is selected, then the player will switch the phone to its full page draw mode, thus increasing the size of the canvas. This would minimize the need for scroll bars as well as show more of the page in a single view. However, usually this means that the labels for the soft key commands will not be drawn, and though the commands themselves will remain available.

If Suppress Scroll Bars is selected, then no scroll bars will draw for this page

#### 3: Player Implementation

The drawing model scrolls the page so that the newly selected object is fully visible.

A vertical scroll bar is displayed if the height size of the page exceeds the current canvas height. A horizontal scroll bar is displayed if the width of the page exceeds the current canvas width.

#### 4: Format Factor Considerations

a: When to Use "Draw Full Page"

Although drawing the Full Page increases the amount of content that will be displayed, on most phones it will also suppress the drawing of the Soft Key Labels. The Soft Keys are still available, but the user will have no visual reference.

This is a trade off in terms of usability which could vary page by page.

b: Conditional "Draw Full Page"

There is a range of form factor heights within both the QVGA and QCIF form factor classes. "Draw Scroll Bars" will deal with all cases, but there is, depending on the page design, phones that will have many pages that will scroll.

An optimal design would be that the page exactly matches the default form factor of the phone, this giving the user the best possible navigation experience. However, a compromise that would reduce the number of required form factor versions would be to use "Conditional Full Page Draw". If this is employed, then for phones whose default canvas height is less than the reference page height for the application, the view is switched automatically to "Draw Full Page" so that the entire page is visible. Otherwise, the optimal visual presentation will occur.

To implement this conditional capability set the page to:

- suppress scroll bars
- draw full page is turned off

When the Player, under these conditions, detects that the default form factor is not sufficient to draw the entire page it will automatically switch to "Draw Full Page".

#### 5: Soft Key Commands

Any number of Soft Key commands can be assigned to a page.

Authoring Soft Key commands will have a similar UI as authoring Lists.

The designer will be able to add, edit and remove a soft key from the Edit Page Dialog Box in a manner similar to that of working with list items in the List Edit dialog box.

The default Back Command will appear and cannot be erased. However, it can be reset to "Goto Page" if desired.

When adding or Editing Soft Key commands the following attributes will be assignable.

- a: A drop down for selection of the attached command. The 8 choices are:
  - · 1. Goto Page
  - 2. Back (this is defined by default)
  - 3. Goto Page & Report
  - 4. Back 1 View & Report
  - 5. Goto Start Page
  - 6. Goto Start Page & Report
  - 7. Login
  - 8. Logout
  - 9. Execute a Web Component

If a command with "Report" is selected, then the normal getParentObjectSelected interface method will be called, with a boolean identifying this as a soft key command. The command name is used as the object name and the command number is used as the object number.

If "Execute a Web Component" is selected, then any Web Component that has been previously assigned to the Page or any object on the Page will become available as a Command Parameter.

b: A text field to define the string that will appear as the soft key list (which, on many phones, may appear as a scrolling pane after a certain number of commands are defined).

### D: Alerts

#### 1: Types

There are 3 classes of Alert Popup Pages, all which will have the same coordinates as regular pages but with a transparent background.

- 1: Application Specific
- 2: Author Specific
- 3: System Default

Alert Pages create a multi-layered drawing model for pages.

The author-defined exit events will display an Alert Popup Page. It is permissible for Alert Pages to call and display another Alert Page.

The Player will first look in the PDL Image Table for the Alert Popup page. If not found it will look at the Author Specific Alert Popup Page Library (much like an Xpressmo Server Page).

System Alerts, there is a default System Alert Template Page available for all applications. This Page has defined text objects for the actual test message that will be displayed based on the real-time condition that triggered the System Alert. The author is able to add other objects as well as control the look of the Text Objects.

There are three types of Popup Alerts.

1: Non-modal

The Alert will draw. All functions for underlying page are active.

A non-modal alert can be discarded by:

- a: A timeout assigned to the Alert Page
- b: An Exit Event
- c: A backend API
- 2: Modal (User dismissible)

The Alert will draw. All functions for underlying page will be disabled except for Exit application. A dim-Page mask will be drawn over the underlying page.

A User Dismissible Modal alert will be discarded on the first "Fire" Key Event.

#### 3: Modal (Event Driven)

The Alert will draw. All functions for underlying page will be disabled except for Exit application. A dim-Page mask will be drawn over the underlying page.

An Event Driven Modal alert can either be discarded by:

- a: An Exit Event
- b A backend API

"Fire" and Author –defined Soft Key Commands are the only interactive events supported. They will apply to the entire Popup Alert object, and it is effective for Popup Alert Type 2. The Events and Animation Tabs of the Resource Director are not available. Navigation within the Alert is not supported. Export and Import of Alerts from the Author and/or System Libraries are supported.

Alert Pages of all three types can be displayed or disposed by backend API's.

2: Authoring UI

When an Alert is created or selected the background will draw as transparent page.

When an Alert is created, there will be a test to make sure that its name is unique, both in terms of other application specific alerts and/ or application pages.

Alerts appear as a choice under "Messages", under the Objects-Messages... menu bar, and in the Layer Inspector as shown to the right.

On the Layer Inspector there is a dropdown showing the current alerts defined for the application. It works like the Pages dropdown.



When an Alert is selected, there will be a list with 3 possible selections.

- 1: Non-modal
  - If "Stay" is selected. A non-modal alert can only be dismissed by:
  - a: A backend API
  - b: A "hide alert" Exit Event.

Otherwise, if not yet dismissed, the timeout will disable the Alert.

- 2: Modal (User dismissible)
- 3: Modal (Event Driven)

These are the same choices that a Page has in terms of defining Soft Key Commands.

The Alert Object Name can be renamed, and, for convenience, the x,y location of the cursor is displayed as well as the size of the underlying page.

The Objects that can be defined for an Alert are Forms (non-interactive), Media, Shapes and/or Text. Child Objects are available. Overlapping Objects are also supported.

- 3: How to make an Alert Page appear
  - 1: By API
    - 2: By Error Condition
    - 3: By Exit Event
      - Slide
      - ListItem
      - Object

### E: Libraries

- a: Publisher Level (publicly available libraries)
  - Pages
  - Page lcons
- b: Author Level
  - Pages (Page Library)
  - Page Icons (all page icons that were uploaded by author)
- c: Application Level
  - Background (Page lcons)
  - lcons (application icons)
  - pages (Express Mobile Server Pages)

### F: Saving Files

The Save As File Menu can be reached either from the File Menu or from the Command Palette.

There are a minimum of six file types that can be selected, and, depending upon the file type, various options will become available. For the case of saving an application, the following is available:

1: Generate QCIF Form Factor

This will be available if the application size is QVGA. (Note that if the application size is VGA then scaling to QVGA will also be available.

2: Embed Application, Images/Slides or Audio Video. 🚰 Save As File Menu 4 **File Name** Generate OVGA Form Factor, NewApp Generate QCIF Form Factor: **File Type** Appleanon \* Embed Application: Current Web Page Embed Images/Slides: Widget Object Application Template Embed Audio/Video: Style Template Reset Web Components: Selected Object Style OK | Cancel Help Fig 2-12

Usually not necessary unless there is interest to make the application fully available in an area that does not have an Internet connection.

3: Reset Web Components.

If the author has manually transferred the application or applications to a different server, then selecting this choice will reset all the Web Components to point to the current server.

File Type can also include the Currently Selected:

- Slide Show
- Complex Vector
- List

### G: Enhanced Navigation Specifcation

#### 1. Overview

"Navigate by Object" navigation, utilizing edge detection, depending upon the page layout, often produced undesirable results. It also imposed various limitations which have now been addressed.

Settings [	*Events 🖗	Animation	🛎 Color 🏝 🕈 Binding	JS.
Web Page I  default	Name Ena	ble Listener	Restore Defaults	
Navigate by	Object		Page Icon Settings Add Page Icon	
UpperLeft	<u> </u>	<u>г</u>		
Page Size	Settings		Remove Page Icon	
OVGA Phon	G 240x260		Navigation Controls	
QVGA Phon	e: 240x320		Soft Key Commands	
QCIF Phone	: 175x185 : 176x208		Master Page Objects	
QCIF Phone QCIF Phone	: 176x220 : 176x144	E	Define Scroll Bar	
•		e caar		Fig

There is now a new button "Navigation Controls" that gives the author the following capabilities.

- Select navigation order identical to that which "Edge Detection" produced.
- Set any navigation sequence for vertical navigation.
- Set any navigation sequence for horizontal navigation.
- For Lists and Slide Shows have controls to have these objects cycle when reaching either end of their elements, or have the focus switch to the next object in the ordered lists.
- Under certain circumstances, when there are no logical conflicts for any navigation event, set the page so that it maintains the focus for all selectable objects simultaneously.

- 2. Authoring UI
  - a. Working with Navigation

Vertical Order	C Horizontal Order one two List three four		
Enable Cycling		three 3 Not Selectable	four [5]
☐ Suppress Navigati	on Restore OK Can	cel Help	

When touching the "Navigation Controls" button, the above dialog box will appear with the current page layout available as feedback.

Only selectable objects will be assigned a navigation sequence. Any non-selectable objects will be drawn but not numbered. The initial state has the "Vertical Order" selected with "Up" and "Down" buttons available. The ordering sequence is both displayed as labels in the feedback and by the object's position in the Vertical List.

The "Restore" button, if touched, will set the navigation order back to the original order as defined by the prior "edge Detection" algorithm.

The feedback is interactive so that objects can be selected for setting various forms of navigation either by:

- 1. Touching an object in the feedback.
- 2. Touching any item in either the Horizontal or Vertical lists.



The graphic above show what would happen if the List Object was touched.

Vertical Order	Horizontal Order	· · · · · · · · · · · · · · · · · · ·		two
one	one			2
List	two		List	
three	Lisi		X	1
rour hue	Inree		13.	
twu	1001 -		÷	
			1	P
	· Left   Right		X	
*			<b>α</b>	
		ļ		four
		L		5
			Nct Selectable	
			L	
Enable Cycling				
<b></b>		• ·		
Sunoress Navigat	DO DOCTOROLOKIC	ancel Hain		

The graphic above shows what would happen if the List item was touched in the Horizontal Order List. Note:

- The "Horizontal Order" radio button was implicitly selected.
- Navigation Control buttons are now labels "Left" and "Right".
- The Feedback now has labels representing the horizontal navigation order.

Chapter 02.21

b. Working with Cycling

The default state for both Lists and Slide Shows is that they cycle. This is show in the graphics above both by a Cycling lcon and the "Enable Cycling" check box selected.



Fig 2-17

Chapter 02.22

Using the "Enable Cycling" check box the Cycling navigation state can be turned on or off. Enable Cycling will also be a check box for both Slide Shows and Lists



c. Changing the Navigation Order

٢

Vertical Order	🦵 Ногі:	contal Ord	ier :	i	<u>or</u>	<u>ie</u>	<b></b>	two	
one	one				Ľ	IJ		4	
LTS)	two		ł				-		
three	List		ł		i –	1	21		
two	three		÷		11	e e	5		
Up Down			•			2e		four (E)	
Up Down			•			e Joi Selectat		four 5	
Up Down		· · · · · · · · · · · · · · · · · · ·	• •			se 3 lot Selectat	de	four (5)	

If the List is selected (see graphic on the top and the "Up" button is touched the result is shown in the graphic below

Vertical Order	C Horizontal Order			two
list	one			4
one	two			
three	LIST	•		1
tour	four	1		
J				
Up Down				
	,		three	tour
				5
				U
			Not Selectable	7
			L	
Enable Cycling			1	
Suppress Navigat	Restorel OK	Cancel Help		

#### d. Navigation Rules with Changing the Page Layout

.

If a page is in the "default navigation" mode, then all operations will proceed and at the time of the page being saved, the current "edge detection" order will be applied.

Otherwise, the following rules will be applied.

If an object is added, the navigation order will be set for that object so that it is at the bottom of both the horizontal and vertical ordered lists.

Note: This would be equivalent to placing the object in the extreme lower right corner of the page if the "edge detection" algorithm was applied.

If an object is deleted it will be removed from the ordered lists. If there is an undo operation, then the object will resume its previously defined place in the ordered lists.

e. "All Objects in Focus" Page

The following page layout conditions must exist for an "All Objects in Focus" mode to be available.

- There are exactly two Selectable Objects in the page. (See appendix D of the User's Guide).
- One object is either a Text Field or a Horizontal Slide Show.
- The other object is either a List or a Vertical Slide Show.

Vertical Order	Horizontal Order	Horizontal Slide Show
Horizontal Slide Sho USI	Horizontal Slide Sho List	×ů×
	• [management]]_management	List
Up Down		
Enable Cycling		
C.C		Autor and a second s

Under the page layout above, the Slide show will accept the horizontal navigation keys and the List will accept the vertical keys. Cycling will automatically be turned on for any Slide Show or List.

FIRE will be honored by the following defaults.

If, as above, there is a Slide Show and List, the List will accept the FIRE event, unless there are no exit events assigned. Otherwise, the Slide Show will accept the FIRE event.

If there is a Vertical Slide Show and a Text Field, the Text Field will accept the FIRE event, unless there are no exit events assigned. Otherwise, the Slide Show will accept the FIRE event.

In the event that the Author selects "Suppress Navigation" for a page, and then, in a subsequent editing event, attempts to either delete, add, or redefine an Object that would cause the necessary conditions for this page to be violated, a warning message will appear. The author will have the choice to continue, and have the "All Objects in Focus" attribute set to false, or cancel the operation.

#### 3. Player Design

a. Navigation Table & Selected Object Initialization

Each page (class WebPageRun or RuntimePageEvents), after it has finished instantiating its contained objects, constructs a table, navTable. This table is assembled from the values of ObjectIntegerAttribute[44] (horizontal order) and ObjectIntegerAttribute[45] (vertical order) of each of the objects, in drawing object order. The table is a two dimensional array defined as:

short[]] navTable = new short[2];

navTable[0] represents the horizontal and navTable[1] represents the vertical.

Each element in the second dimension represents the object navigation index (nav index). The index into the 2<sup>nd</sup> dimension is the object's drawing order. The size of the 2<sup>nd</sup> dimension is the number of objects in the page, including objects that are not selectable.

navTable is cleared on end of page and allocated on start of page.

Focus means that an object has control on the page. An object in focus normally gets first right of refusal for all key and pointer events. In UI, an object in focus is often, but not always, indicated with a visual difference from other objects on the page, e.g., by change in color, shape or size..

A page contains a reference to the currently selected object, called pObjSelected. A page where no object has focus has pObjSelected set to null. See end of section 5.2 Basic Algorithm for a more detailed description of how pObjSelected is initialized.

#### b. Basic Algorithm

In the following description the notation  $\{x \mid y\}$  means use x when processing horizontal navigation action and y when processing vertical navigation action.

"nav index" is the value stored in navTable that indicates the position of this object in a horizontal or vertical navigation.

After navTable is constructed, and pObjSelected is set, when the page receives a navigation event (e.g., pressing or selecting the left, right, up, or down arrow key) that is not captured by the current object, the following occurs:

- 1. Obtain the {horizontal | vertical} nav index of the object referred to by pObjSelected from pObjSelected. lected.ObjectIntegerAttribute[{44 | 45]].
  - a. If the event is navigate {left | up}, find the value in navTable[{0 | 1}] that is 1 less than the current object's nav index.
    - i. The new object's draw index is the index of the value found in navTable.
    - ii. If the current object's nav index is 0, the position and focus remain unchanged.
    - iii. If pObjSelected is null, the new position is the object which has nav index 0. If no such object exists, no change in focus is made.

- b. If the action is navigate {right | down}, find the value in navTable[{0 | 1}] that is 1 more than the current object's index.
  - i. If the current object's nav index is for the last position in the page, the position and focus remain unchanged.
  - ii. If pObjSelected is null, the new position is the object with nav index 0. If no such object exists, no change in focus is made.
- Set pObjSelected to the object whose position in the drawing order is the index into navTable{{0 | 1}} where value derived in 1a and 1b above resides.

Note: the initial value of pObjSelected is set by one of the following means:

- a. If there is a value in the CallPoint object that indicates an object index, set the initial pObjSelected to the object whose drawing order is the value stored in CallPoint.
- b. If there is no value in CallPoint indicating starting object (indicated by -1), then use the object whose value in navTable[0][] is 0 in the horizontal navigation chain as the pObjSelected reference.
- c. In case 2, if no element in navTable[0] contains 0, set pObjSelected to null.

#### c. Degenerate Cases

If there are no objects on the page, navTable will have a dimension of [2][0]; pObjSelected will remain null while this page is active.

If there are no objects that are selectable, navTable will have a dimension of [2][0]; pObjSelected will remain null while this page is active.

If there is a mixture of selectable and non-selectable objects on the page, the slots in navTable that represent non-selectable objects contain -1. No slot with value -1 can be selected as the next selectable object. > This implies that ObjectIntegerAttribute[44] and ObjectIntegerAttribute[45] must also store -1 for objects that are not selectable, as navTable is populated from these attributes.

#### d. Illustrative Example

Let us say that the object on the left have natural drawing order, starting with top left element and proceeding in a left-to-right then top-to-bottom order. Assume the horizontal movement, as numbered inside each object in the illustration to the right. Then navTable for this page contains the following values (name in parenthesis is object name):





Fig 2-23

Observe the following:

- The navTable position corresponding to object "one" has 0 to indicate it as the first element in the horizontal navigation chain. A left navigation when focus is on "one" does not modify the focus.
- The object "four" is the last horizontally navigable object. A right navigation when focus is on "four" does not modify the focus.
- The object "Not Selectable" has -1 to indicate that is not a selectable object as far as focus is concerned.
- The number of elements in navTable[0] is equal to the number of objects on the page. In this example, it is 6.
- In the illustration above, pObjSelected refers to object "List", and its horizontal nav index is 1. Moving left changes focus to object "one", and moving right changes focus to object "three".

- 4. PDL Specification for Enhanced Navigation
  - a. Page Definition

BooleanAttribute[8] =	Default Navigation Overwritten
BooleanAttribute[9] =	Suppress Navigation

b. Primitive Object Definitions

ObjectBooleanAttribute[ 28]= SlideShow & List: Suppress Cycling

ObjectIntegerAttribute[44] = Horizontal Navigation Order (-1=not selectable, 0 =first, 1...n-1 = additional order, where n is number of selectable objects on page)

ObjectIntegerAttribute[45] = Vertical Navigation Order (-1=not selectable, 0 =first, 1..n = additional order, where n is number of selectable objects on page)

. •

.

xpressmo.com

IPR2021-01146 Page 00250

EXPRESS MOBILE User's Guide

Chapter 03

Introduction to Objects

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 3: Introduction to Objects

.

A:	How to Create	03.05
B:	Keyboard Operations with UI Objects	03.05
C:	Persistent Variables	03.06
D:	Layout by Pixel vs Layout by Font	03.17
E:	Suppression of Selection Rectangle	03.19
F:	Using Object Libraries	03.20
## Chapter 3: Glossary

Objects TAn intelligent entity that has both a defined appearance and a set of defined behaviors. These behaviors can be activated based on: 1: User Interaction 2: Time 3: Location when in an animation. Suppression The selective disabling of a certain function of behavior. Examples include the suppression of the Selection Rectangle for a given UI Parent Object or the suppression of one or more objects that are defined in the Master Page for a given Application Page. Layout by Pixel Overrides the changes to the width and height of a Text Object, that would normally occur based on the Font Metrics of the mobile device. This guarantees that the size and position of all text objects remain unchanged across device types, although the actual size of the text itself will now vary to a greater degree.

# Introduction to Objects

Edit		Navigation			Media	Shapes	T.
Goto Page						·	Fig 3-1
Copy Page Page Viewer		The comm	ands for wo	rking with	Objects	Events	Page
Navigation	•	four sourc	es:	tollowing	Navig	ation	•
Messages	•	1.The Ob	ject Orient	ed Com-	Mess	ages	•
Shapes		2.The Pop	m the looida Sud Menu t	r above.	Shape	9S	
Photo		which app right mou	pears by u se button v	using the when over	Forms	3	•
Video		the Canvas	s.		Photo	• • •	
Slide Show		3.The Ob right.	jects Meni	u to the	Slide	Show	
SlideShow Library		4.The Ob	iect Icons	from the			
Virtual Tour		"Pages/Ot	ojects Panel	' below.	Text	An an An An An	•
Virtual Library					Variat	les	•
Mobile GIS Service		81		1			
Button		Objects	over			Ungroup	
Paragraph		19: Traffic_	_image r image				Fig 3-3
Variables	•	17: Weather 18: Traffic	erDetail textarea	, n			
Forms	. •	14: gnews( 13: gnews(	)1 )2				
Select All		12: gnews( 11: gnews)	)3 )4 )hiect 17				
Deselect All				≥ 1			
	Fig 3-1	l		Fig 3-4			

Chapter 03.5

### A: How to Create

There are three different ways to create UI Objects.

- 1. The Object Oriented Commands from the Toolbar.
- 2. The Popup Menu, using the right mouse button when over the Canvas, and no object is selected.
- 3. The Objects Menu in the Menu Bar.

## B: Keyboard Operations with UI Objects

1: Keyboard Shortcuts

The following shortcuts are available with the Publisher:

- a: New\_ (CTRL N) (COMMAND N) for Macintosh
- b. Open\_ (CTRL O) (COMMAND O)
- c. Save\_(CTRL S) (COMMAND S)
- d. Cut\_ (CTRL X) (COMMAND X)
- e. Copy\_(CTRL C) (COMMAND C)
- f. Paste\_(CTRL V) (COMMAND V)
- g. Delete\_(DEL)
- h. Deselect\_(CTRL D) (COMMAND X) Deselects all objects on the page.
- i. Undo\_ (CTRL Z) (COMMAND Z) Currently this is an undelete only function.
- j. Edit\_(CTRL E) (COMMAND E) Opens up the Edit Dialog Box
- k. Insert\_ (CTRL I) (COMMAND I) Inserts a new Page
- 1. Restore\_(CTRL R) (COMMAND R) For Images only. Will restore the image to its actual size.
- 2: Resizing with the Keyboard.

For images and video, with the Shift Key pressed, touching the arrow keys will cause the Object's width or height to increase (right or down arrow keys) or decrease (left or top arrow keys).

3: Dragging with the Keyboard.

With the CTRL key pressed, touching the arrow keys will move the selected object in the direction of the arrow keys.

4: Maintaining the Aspect Ratio.

The aspect ratio of the selected object can be maintained by selecting an attachment point with the mouse and then, with the Shift key pressed, dragging the mouse.

## C. Player Data Persistence Specification

### 1. Using Escape Characters

In the player code, there is an existing implementation that allows storage and retrieval of data objects required by an application. The data is stored as a key-value pair, much like in a properties file. The data persists across sessions, and is only removed when the user deletes the application/player from the device. The current limitation is that all keys and values must be a String object. This is suitable for storing text and numbers but is not suitable for storing complex objects, such as all the attributes of a list item.

Data persistence is accessible in the player, and can be designated by the author of the application using a special notation in text fields. %variableName% is the format, where the variable escape "%" characters act to enclose the variable name. A typical use of this is to ask the user for a login name and have this login name persist within the application, and across sessions.

The %variableName% can be placed in any text field where normal text is placed. The player will automatically substitute the key, "variableName" with the value associated with that variable within the persistent storage module. If variableName is not found in persistent storage, then a null value is returned.

The population of persistent storage was either accomplished with an Exit Event ("Set Starting Page") or implicitly with Chat for the "UserName" Handle.

- 2. Supported Objects
  - a. Scalar Variables

The following objects will be supported for Player Data Persistence for Scalar type Variables:

- Text Fields
- Text Areas
- Choice Objects
- List Objects
- Check Boxes
- Video Objects
- Launch Strips

Object Type	Value that is saved
Checkbox	State of the checkbox (true = "checked", false = "unchecked")
Dropdown List	Last selected item's display text
Complex List	Last selected item's display text[0] (top left text)
Text Field	String value of text field
Text Area	String value of text area.
Video	The video URL. If DefineByURL then it's the absolute URL
Launch Strip	The currently selected channel

The following objects will be supported for Player Data Persistence based on a comparison to a hidden String Attribute:

- List Objects
- Slide Show Objects
- Text Button
- Vector
- Image
- .

All Scalar type Variables will support a replace only function.

b. Non-Scalar Variables

The following objects will be supported for adding new entries to Non-Scalar Variables:

- Text Fields
- Text Areas
- Check Boxes

Object Type	Value that is saved
Checkbox	State of the checkbox (true = "checked", false = "unchecked")
Text Field	String value of text field
Text Area	String value of text area.

The following objects will be supported for selecting the default value of a Non-Scalar Variable:

- Choice Objects
- List Objects

- 3. Persistent Variable Types
  - a. Scalar:

Used when the variable will only have a single value. Adding a value to a scalar variable is equivalent to replacing the existing value with a new value.

- b. Non-Scalar:
  - Stack (LIFO):

As each value is placed in the persistent variable record, it is placed on the top so that, when retrieved, they appear in the order based on the newest item first. The list of values must be unique

Vector (FIFO):

As each value is placed in the persistent variable record, it is appended so that, when retrieved, they appear in the order based on the oldest item first. The list of values must be unique

• History:

For adding values to the persistent variable record, History will act similarly to a Stack, but will not enforce uniqueness.

All persistent variable types will be supported for all objects as defined in section 3 **Supported Objects** in terms of adding values to the persistent variable record. However, only the following objects will display the full Stack, Queue or History and permit the override of the default selection.

- a. Choice Objects
- b. List Objects

If no selection has yet occurred, then all other objects will convert the multi-item record to a scalar value based on the selection as follows:

- a. Stack: Top Item
- b. Queue: Last Item
- c. History: Top Item

### 4. User Interface

When a supported Object is selected, the initial state will be two available checkboxes:

- "Remember Me"
- "Suppress Selection Rectangle"

Settings Events	Animation	- Col	orce	Bin	
Define the Text Field St	ring	F	Object Sty		
Font Name TimesRoman_+	Font Effecte Regular	Ē	Font	Size	
Object Name Text Field_1	Frame Style		Fran 1	ne Wir	
Left Top Width	Height	Lock	Hide		
Attributes Draw Default 🚽 🔽	Frame	Selecta	able	Edi F	
Remember Me	Suppress S	election	Recta	ngle	
Add Persistent Variable					
1061 -		2			
				Fig 3-5	

When the "Remember Me" attribute is selected, then two additional UI objects become enabled.

### a. Add Persistent Variable

The default variable name will be "PageName"\_"ObjectName". However the author can rename it to any name they wish. However, if the name, as entered, has already been defined the text will turn red. If OK is touched before correction then an Error Message will appear.

Add Persistent Variable 3	
Persistent Variable Name	Data Structure Type
street	History List -
	Scalar
	Stack (LIFO)
	Vector (FIFO)
	HEORYLE) N

Fig **3-6** 

If an object is selected, and it is a then there are four types of data structures that can be assigned as described above.

Upon a successful completion the Variable name will be added to the drop down menu and the current object will have that variable selected.

### b. Select a Persistent Variable

At any time the author can choose a different persistent variable to assign to the selected object. This means that multiple objects on the same page and/or on different pages can share the same persistent variable.

Remember Me	Suppress Selection Rectangle
Add Persistent Variable	
foo1 💌	
foo1	
1002 1- 1 19	

Fig 3-7

c. Working with History Lists.

To create a History List, first create a Dropdown Menu and then select the "Dropdown History List" in the List Selector Type choice list. A "Select History Variables" button will appear for assigning non-scalar variable lists to up to three different elements for each item in the History List.

Touching this button will display the following Dialog Box.

Settings	Events	Animation	Color - Bindings
List Visible It	ems	Draw Frame	Object Style Default
Font Name TimesRom	an 💌	Font Effects Regular 💌	Font Size
Object Nam Dropdown	ie Object_1	Frame Style	Frame Width
Left To 10 30	Width	Height Lo	ck Hide
DropDown	Items	List Selector	г Туре
Add Items	<b>-</b>	Dropslowm Select Hist	tory Variables

All History Persistent Variables that have been defined on the current page will be displayed as candidates. Any other reference to History Variables from objects that are not on the same page that the "History List" resides will be display only, and will be set as non-editable.

In the example to the right there were two text fields that had been assigned "history" persistent variables "street" and "zipcode" respectively.

The graphic to the right shows the result when selecting the "street" candidate and touching the "Add" button. Note that the "Remove" button is now enabled.

The graphic to the right shows the result when selecting the "zipcode" candidate and touching the "Add" button. Note that a comma and space has been added as delimiters.

The Text Field is interactive so that touching "street" will select the "street" string as well as the "street" item in the candidate list. Alternatively the "street" item could be selected directly by touching the "street" item in the Candidate List.

Finally the graphic to the right shows the result when touching the "Remove" button when the "zipcode" candidate is selected.



xpressmo.com

d. Removing a Persistent Variable.

If "None" is selected for an object that had a previously assigned persistent variable, then its reference is removed. If there are no remaining references to that persistent variable, then the persistent variable is also removed.

e. Removing all Persistent Variables in an Application.

If the author wishes to clear out all persistent variables defined in an application, then Select "Remove-all Remember Me" from the Edit Menu.

- f. Behaviors
  - 1. Objects with the scalar persistent variable attribute can serve a dual purpose:
    - Setting a Persistent Storage Value.
    - Retrieving and Displaying a previously stored Persistent Value.
  - 2. If there is no interest in having that object reset the current stored value for that persistent variable then give that object either:
    - Not-editable attribute
    - Not selectable attribute
    - Text Buttons, Vectors and Images only retrieve a stored Persistent Value
  - 3. The memory for persistent variables is application specific. Even if there are two applications with the same name, they will be differentiated based on the UserName attribute.
  - 4. FIRE will set the persistent variable's value as described in "Section D.3.b" on the previous page. In the event that there are any objects on the same page that share the same persistent variable, then they will be updated immediately. If one of these objects is a check box, then the checkbox will become selected if, and only if, the string of the object that is fired on equals "true".
  - 5. Any Exit Event on the page that changes the page view or the application will implicitly set the persistent variable's value for all objects on that page that have the "Remember Me" attribute as described in "Section C.3.a" on the previous page.
  - 6. The rules for the current %variableName% notation for Text Fields will be augmented so that, as long as there are two, and only two variable escape "%" characters in the String, then:
    - · Embedded spaces will be permitted.

•	The	%v	ariableN	lame%	substr	ing will	be	replaced	•	by	its	val-
	ue.	Any	text	outside	the	%variableNa	ne%	substring	will	be	pres	erved.

- 7. If the persistent Variable is other than Scalar, and the rules as defined in Section 4 Persistent Variable Types above apply.
- 8. Non-Scalar History variables will be editable only on the page that has a History List object defined. When there is an exit from such a page, all history variables that are referred to by the History List will be simultaneously updated with their current value, only if there is uniqueness as a group of values.

Chapter 03.14

### **Publisher Interface**

#### g. Persistent Variable Naming Conventions

Persistent variables are all key-value pairs. The key name identifies the variable and the value is the stored information associated with the key in String form. The VariableName follows the convention:

ApplicationAuthorAccount.ApplicationName.VariableName.

h. Player-side Implementation

On the player side, the following will be assumed:

- 1. For persistent variables used at the object level, both scalar and non-scalar variable types are supported.
- 2. For persistent variables used anywhere else, including at the page level, only scalar variable types are supported.
  - i. Present Situation

The underlying implementation for persistent variables uses Record Store that is available on all MIDP 2.0 devices. Each variable is stored as a single record in RecordStore. All variables are essentially of type String, organized as a key-value pair, in the following format:

key=value

"key" is the variable name, and "value" is the String value. RecordStore stores byte arrays only, so the String is converted to a byte array before storage and converted from byte array to String upon retrieval. Only items that can be converted into a String type can be stored as persistent variables. Once the key-value pair is constructed it is stored in RecordStore.

When a value for a variable is requested, the persistent variable Record Store is scanned for the key matching the requested variable's name. When one is matched, the value portion of the record is extracted and returned. If no record has a key with the named variable's name, null is returned.

ii. Proposed Implementation

In the proposed implementation, each variable will have as its value an array of String elements, or in Java notation, String]. This is the case even for scalar variables – it will be stored as a single element array, whose size is 1. The String] structure requires serialization into a byte array. There will be serialize() and deserialize() methods provided to handle this. In addition, the ability to store a variable number of metadata associated with the variable is implemented. The proposed storage format for variables is as follows:

<key>=<header>\000<elementAt\_0\_value>\000<elementAt\_1\_value>\000 ... elementAt\_n-1\_value\000

"<key>" is the same as before; it is the <u>fully qualified</u> variable name.

"=" is the same as before; it is the separator between the key and the value.

<header> consists of metadata parameters for the variable. Its syntax is a semicolon-separated list of key-value pairs. The last key-value pair is also terminated by a semicolon. As the semicolon character is used as a delimiter, it is not allowed in the values for header key-value pairs. The order of appearance of the metadata is not significant. The number of parameters is variable. This allows for easy modification of the number of metadata parameters that can be recorded in the header.

type=n;maxElements=nnn;nElements=mmm;selectedIndices=a,b,...,n;

type is the variable type. This parameter is required. The valid values are:

0: scalar - can have at most one element as its value

- 1: LIFO can have multiple elements as its value. Behaves like a stack. Uniqueness enforced (no to value elements can have the same value).
- 2: FIFO can have multiple elements as its value. Behaves list a queue. Uniqueness enforced.
- 3: LIFO with uniqueness not enforced (there can be multiple value elements with the same value).
- 4: Sorted List can have multiple elements as its value. List is sorted according to the default collating sequence. Uniqueness enforced.

maxElements is the maximum size to which this array can grow to. For scalar variables, this value is 1. For all other non-scalar types, this value is 127. This parameter is required.

nElements is the current number of elements stored in the variable. This number must be less than or equal to maxElements. This parameter is required.

selectedIndices are one or more indices of the items that are selected. Each value is an index into the array that follows. Multiple indices are separated by commas. This parameter is optional.

"\000" is the field delimiter within the value portion of the record. It delimits between the header and the vector elements and between vector elements. The last value element must also be succeeded by \000.

<elementAt\_x\_value> is the value of an array element, where x is the index of that element within the array. Each <elementAt\_x\_value> must be a UTF-8 encoded String and must not contain the NUL character.

iii. Implementation Notes

PlayerPersistentVariables will be augmented, such that setVariableXXX has a type parameter, either explicitly, for those methods that handle all types, or implicitly for those that handle only scalar variable. There will be overloaded methods in setVariableXXX() and getVariableXXX() to store and retrieve non-scalar variables.

Two new methods to get and set the selected items attribute, getSelected() and setSelected(), will be implemented, such that one can record and retrieve which items within a list are in the "selected" state. This would apply to all variables including scalar.

To manage the addition of value elements to a variable, rather than replacement of the value, a new method addElement() is created. Depending on the type, the new element will be added as the 1<sup>st</sup> element (LIFO, History), last element (FIFO), or in lexicographically sorted order (Sorted List). If addElement() is called on a scalar variable, the existing value is replaced with the new value, equivalent to calling setVariable().

### i. PDL Specifications

Page Definition

BooleanAttribute[14] = Page has a History List Object

Primitive Object Definitions

ObjectIntegerAttribute[43] is the variable type associated with the object

ObjectString [10] = First Persistent Variable Assigned at History List Item Level

ObjectString [9] = Second Persistent Variable Assigned at History List Item Level

ObjectString [8] = Third Persistent Variable Assigned at History List Item Level

## D: Layout by Pixel vs. Layout by Font

### 1: Background

All Form objects currently are by, default, sized in terms of height, by the line height as generated by the selected font size. Submit, and Clear buttons have their width defined by the width of the character string, as also determined by the selected font metrics.

Although there is a provision to "don't scale" in many cases, it still means that the Author must control the actual layout through the font metric settings.

The difficulty is that MIDP does not have a font generator, and the fonts that are supported on phones can have very different font metrics, for the same logical font size.

Since all other platforms do have compatible font metrics, the current method will remain, but there will be a new selection under the "Project" menu, "Define Form Objects by Pixels". This will become a user preference and will be remembered between editing sessions. It can be changed at any time, even while editing an application.

Froject	Objects	Events	Pages
Login F	Required		
✓ Genera	te Univer	sal Playe	r
≪Definie	FormObj	ecis (iv Pi	i ələ
Genera	ite Server	Pages	
✓ Activate	Server A	Pls	· · · ·
Extend	Player AP	Is	
			Cia 21

#### Fig 3-16

#### 2: UI Implementation

Font Size will be a text field showing the calculated value and will not be selectable

Both the width and height fields will always be selectable. The height field will still represent the total height of the Object. For Text Areas and Lists the Line Height will always be the Object Height divided by the Number of Visible Lines.

In addition, all Form objects, other than a check box, will now have 8 drag points so that the object can be resized with the mouse. Upon releasing the mouse a new font, on a best fit basis, will be selected and the text will then be reformatted

Settings Events Animation Color Bindings **List Object Visible Items Object Style** 4 - **X** Default . Font Name Font Effects Font Size TimesRoman 👻 Regular 16 👻 **Object Name** Frame Width Frame Style List Object\_3 None -• 11 Hide Restore Default Left Top Width Height Lock 78 102 150 88 Γ. r Γ. Fig 3-17

based on this new font. Only a change in the height will affect the font, a width change will only change the line lengths that will be drawn.

For Submit and Clear buttons there will remain a "Don't Scale" Check Box. If scaling is enabled then the width of the button will be dynamically calculated on the phone based on the actual length of the Text String after a best-fit font is selected.



Fig 3-18

### 3: Effect on Vertical Positioning

All text objects, including text buttons and paragraphs will be affected.

The only exceptions are Lists as they were previously vertically aligned by the top of the bounding box of the first item in the list.

When in "Define Form Objects by Pixels" mode the following objects will be vertically aligned based on the top of the bounding box of the first text string.

- Paragraph
- Text Button
- Submit Button
- Clear button
- Text Field
- Text Area
- Drop Down (Choice) List.

When in the normal "Font Typographic" mode these objects will be vertically aligned on the baseline of the first text string.

## E: Suppression of Selection Rectangle

Normally a selection rectangle will be drawn around any "landable" object as defined in Appendix D. The primary color can be set under the Color Tab when the "Master Page" is selected. The player will use this color to draw a 2 pixel wide 3d rectangle with the color variations selected as if the sun was coming from the upper left. (The algorithm is identical to that of Java AWT draw3DRectangle() api.

If the "Suppress Selection rectangle" checkbox is selected, then the object will still be selected but no selection rectangle will be drawn.

Joiz .	Setting	s Ev	ents' i	Animatio	n 🕴 Co	lor 🧤	Bin
	Define	the Text	Field Str	ing	— j	Objec Default	ct Sty
	Font Na	ame		Font Effe	cts	Font	Size
	Times	Roman	•	Regular	•	16	⊡
	Object	Name		Frame St	yle	Fram	e Wic
	Text Fi	ield <b>_1</b>		None	<u> </u>	1	<b>-</b> ]
	Left 10	Top 110	Width 100	Height	Lock	Hide T	
	Attribu	tes	Draw	Frame	Select	able	Edi
	Defaul	t	ন -		2		Ŀ
R	ememt	er Me	A t	Suppress	Selection	Rectar	ngle
<u>A</u>	dd Per	sistent V	ariable				
ſſ	001		-				

Fig 3-19

## F: Using Object Libraries

### 1: Saving

The following objects can be placed into a private library for use with any application:

- a. Slide Shows
- b. Lists
- c. Vectors
- d. Images
- e. Video

Images and Video, if they were originally resident on your local system, will automatically be placed in their respective libraries when an application is saved.

For Slide Shows, Lists or Vectors, be sure that the object is selected and then choose "Save As.." from the File Menu or touch the Save icon from the Command Floating Palette. Under File Type a choice will appear for saving the selected object.

All of these five object libraries will also have public versions. See your System Administrator for adding any of your private content into these public libraries if you want them available to all other Authors.

2: Retrieving

A standard way to retrieve object libraries is from the Object Libraries menu under the File menu

FIC ED	t View	Project	Object	s Events	Pa
New Open	;	S₩+N ▶	•	90. Marsaa	• [
Close Quit Save Save As		Cirl+9			
Import F Import A Import Y	Page Nort Vidget			•	
Master I	lenmos. °age	<u> </u>	lingoni Import Import	hileges. Video SlideShov	
Copy Ar	plication	•	Import Import	Vector Lişt	
		~			Fig 3-2

There are also contextual ways to work with object libraries as described below.

a. Slide Shows

Slide Shows can be imported into an application in the following ways:

- 1) Right Click Mouse and Select Slide Show Libraries:
- 2) Work with Slides and "Enter Slide Library" from the Import Icon.
- b. Lists

"Import List ... " from the File Menu.

c. Vectors

Vectors can be imported into an application in the following ways:

- 1) Right Click Mouse and Select Shape Libraries.
- 2) Shapes Library from the Shapes Icon from the Icon bar.
- 3) Touch "Enter Shapes Library" from ShapesÖ from Objects in the Menu Bar.
- d. Images

Images can be imported into an application in the following ways:

- 1) Picture from the Media Icon from the Icon bar
- 2) Touch the "Replace Photo File" button in the Resource Inspector.
- e. Video

Video can be imported into an application in the following ways:

- 3) Video from the Media Icon from the Icon bar
- 4) Touch the "Replace Video" button in the Resource Inspector.

,



© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 4: Working with Navigation Objects

A:	Creating a Launch Strip	04.04
B:	Working with Launch Strips	04.04
C:	Working with Complex Lists	04.07
D:	Inter-Application Navigation	04.08
E:	Bubble Help Color	04.08

## Chapter 4: Glossary

Uber Launch StripThis is the Launch Strip assigned to the Initial or Uber-Application. It<br/>will be available for any application page for the Uber-Application or<br/>any Sub-Application.Explode EffectThe visual behavior associated with a launch strip as set by the<br/>author, when there is a change in focus to a lcon.Bubble HelpThe optional text string that can be defined for each lcon in a launch

strip.

## A: Creating an Launch Strip

There are 2 starting points for Launch Strips as shown to the right. Once created, there are no restrictions in terms of working with these attributes.

The only differences in these two choices are:

- 1: Orientation
- 2: Type of Launch Strip

You can also add a slide transition and auto sound when navigating between lcons.

It is possible to have both a vertical and horizontal Launch strip defined simultaneously.

## B: Working with Launch Strips

Use the Resource Inspector to define the Launch Strip.

1: Maximum Number of Visible Slides

This setting will display, in the orientation selected, the number of actual slides, or the Visible Slides setting, whichever is smaller.

2: Explode Effect

Determines the visual effect of a slide being selected.

Explode Arc and Explode Icon are supported

3: Vertical

Switches orientation.

4: Explode Percent.

This defines the relationship between the Selected Icon Size and that of the smallest Icon. For Exploding Arc, the intermediate icons are the average between the two.



Settings Events	Animation	Color Bindings	].
Max. Visible Slides	Explode Effect	Object Style Default	
Vertical Explode (%)	Scroll Bar Width	Pixel Separator	
Object Name HorizontaliconStrip	Frame Style	Frame Width	
Left Top Widtl 56 242 90	Height Loo 18	ck Hide Dim Page	
Select Scroll Bar Posi Not Visible	tion Wor	k With Icons	
Horizontal Icon Strip S Bottorn: Pans Up	etting Vertic	cal Icon Strip Setting ert to Slide Show -	
Horizontal Active Icon	s Verti	Cal Active Icons	
Chat Room Status	Fram ]	e Delay(milliseconds)	
Chat Room Settings Chat Loading String	Numt	2 -	
ſ	- Reme	mber Channels	
Chat View Only	Suppre	ss Selection Rectangle	
		Fig 4	

rig 4-2

xpressmo.com

### IPR2021-01146 Page 00276

5: Scroll Bar Width and Select Scroll Bar Position

These settings control the presence, width, and placement for a Scroll Bar.

6: Pixel Separator:

This is the number of pixels, from 0 to 15, that will define the separation between slides.

7: ObjectName

This name is fixed and is not editable.

8: Frame Style and Width:

Not used with Launch Strip.

9: Left, Top, Width and Height.

These values should generally not be changed, as the positioning and size is being calculated automatically by the number of visible slides, the size of the original images, the Explode Percentage Setting, the Exploding Effect, the Page Size, and the Orientation and placement selection. However, to adjust the value that controls centering from "centered within the page" to something else based on aesthetics is available.

10: Lock and Hide.

Not used with Launch Strip.

11: Dim Page

This determines whether the page should be dimmed with the Launch Strip is visible and active.

12: Work With Icons

This permits the insertion, deletion, and setting attributes for specific icons.

13: Setting Navigation for Starting an Application on a Specific Page

If the Execute Application Exit Event is chosen, then both a dropdown list for the available applications in your workspace will become available for selection, as well as a text field that lets you determine the page name that you want to be the initial page.

14: Horizontal Launch Strip Setting (or Vertical Launch Strip Setting)

This determines where the Launch strip will be placed based on its orientation, whether you want it always visible or enter from a page edge, or whether you want this Launch strip to become a slide show.

Bubble Help		
catl	₩.	
Select Application	Starting Page	
2.0 💽 10 💌	Application	
ransition(Sec.) FPS	Select Exit Mode.	
02_catico.png	Replace Photo	
Media File Name.		
Unassigned	Define RSS Source	
RSS Feed.		
Г		
Enable Mouse.	: IconStrip	
Stay - None		
Delay(Sec.) Tra	insition Animation.	



### 15: Horizontal Active Icons (or Vertical Active Icons)

This sets which lcons, based on the number selected and their order, will immediately be available for selection. Moving icons from a selected to/from an unselected state is managed by the lcon Selection Complex List object.

16: Frame Delay and Number of Frames

These settings are only available for the "Exploding Icon Type". This sets the speed and granularity for the Explosion and transition effect for this type of Launch Strip.

17: Remember Channels

If selected all changes to the active channels, as controlled by a "Personalize Channel" List Object, "Activate Channels" Choice Object, or "Deactivate Channels" Choice Object, will be remembered between sessions. This only applies to the Uber Launch Strip.

18: Suppress Selection Rectangle

The selection rectangle will be suppressed if selected.

19: Selective Suppression of Launch Strip Availability

It is possible, at the page level, to selectively disable either the Vertical Launch Strip, the Horizontal Launch Strip, or both

Web Page Name Enable Liste default T	ener Anticipatory Streaming
Navigate by Object	Page Icon Settings
Page Size Settings	Remove Page Icon
Browser Window QVGA Phone: 240x260	Soft Key Commands
QVGA Phone: 240x320 QCIF Phone: 176x186 QCIF Phone: 176x208	Suppress Horizontal Icon Strip
Treo 600: 165x145 Treo 650: 320x320 VGA Standard: 400x600	Suppress Vertical Icon Strip
VGA Standard: 480x600 VGA Landscape: 480x640 Std Landscape: 800x600	\$
Std Portrait: 600x800 Custom Size: 2000x8000	

Chapter 04.7

:

## C: Working with Complex Lists

Every Item in a Complex list can be assigned a navigation event. The List to the right shows some of the choices. The chapter on Events describes each of these Navigation Choices.

	Item Events and Web Services	· · ·
Goto	External Web Page Launched in a Nev	w Wind
Goto	a specific Page View	
Goto	External Web Page replacing the Top	Frame
Goto	the next Page View	
Exec	ute Java Script Method	
Paus	e/Resume Page Timeout	LS
Exec	ute an Application	
Goto	a Specific Slide in a Page View	
Exit A	pplication	
ExitF	layer	
Place	Phone Call	
Send	String on FIRE	-
٠ 📾		) <b>&gt;</b>
		Fig 4-5

Each Item in the Navigation List can have up to two lines, and each line can have up to 2 text strings and or two image icons. The contents for these text strings and image icons can be assigned either by the author or, through Web Components, by a Web service.

Settings Events	Animation C	olor Bindings
Object Attributes *	List Attributes	Item Attributes
List items	List Styles	List item Layout
49ers win big! 🚽	default 💽	1st Line: 2nd Line:
Add item	Leading	
Remove Item	Line	Title: 🔽 🔽
Remove All	Use Alt.	Font:
List Selector Type	Body	icon: 🗂 👘 Г
Multiple Select	Body S	tring: <b>Г</b>
Multiple Select	Use Alt.	Font:
Category List		
Personalize Channels	Right Ju	istify:
Select widgets from Us	Se	o. Alternate Cant Chiday
Phone List SMS Liet	Brief	
Select Widgets from Ri		
Phone List from PIM	cond Line Indent	
	0 💌	Define Second Color
Unselected Item Icon		

Fig 4-6

## D: Inter-Application Navigation

Similar to B:13 above, it is now possible to assign a particular starting page when utilizing the ExitEvents for all XpressMo Objects and from a particular Choice Object Item or a Complex List item.

There are up to 2 Uber Launch Strips that can be assigned during a Player Session, a Vertical Uber Launch Strip and/or a Horizontal Uber Launch Strip.

The first Uber Launch Strip-enabled application that is visited by the Player during a given session will be the owner of its Uber Launch Strip. This could be the Uber-Portal, or, assuming that the Uber Portal only has one Uber Launch Strip defined, then the first application that defines the other type of Uber Launch Strip will be the Owner.

Ownership as defined as having a set of Icons that remains persistent throughout the session. These Icons may be visible or invisible, but they will be persistent.

When a subordinate application is visited that has an Uber Launch Strip whose type already has an owner, then the subordinate applications loons will be appended to the Uber Launch Strip. This process can continue for any number of Application Layers.

When the owner of an Uber Launch Strip is visited, all subordinate icons are immediately removed.

## E: Bubble Help Color

There are 3 color controls for bubble help.

1: Bubble help background:

This will share the same color as the slide background color.

2: Bubble Help Text Color:

This will be the same for all bubble help

3: Bubble Help Frame color

This will be the same for all bubble help



Fig 4-7

.

EXPRESS MOBILE User's Guide Chapter 05 Working with Message Objects

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 5: Working with Message Objects

A:	Working with Chat	5.04
B:	Working with Phone Calls	5.07
C:	Working with SMS Messages	5.09

D: Working with Rich Media Alerts 5.11

.

# Chapter 5: Glossary

PIM Data

**Rich Media Alerts** 

The data available on the mobile device such as Contact Lists, TODO Lists, and Event Lists.

Alert Pages that can contain both static and dynamic content. These pages can be sent to one or more users that have the Express Mobile Player installed.

Chapter 05.4

## A: Working with Chat

### Setting Colors:

Chats add two additional elements to a Slide Show. The first is the Chat Trail itself and the second is a Text Edit Field where anyone you authorize can add time and date stamped messages to your Chat Trail. Once added, anyone who visits this Chat Application, and navigates to the particular slide that the message was added to, will be able to read that message, and through manual and/or automatic scrolling, view all messages that were added to this Chat Trail, in chronological or reverse chronological order.

To help add your own look and feel to your Chat you can set the text and background colors for both the scrollable Chat Trail as well as for the Editable Text Field.



rig 2

### Setting the Chat:

- a: Chat Status: This control gives you the ability to determine where you want the Chat to be placed relative to the slide show.
- b: Chat Database: Publisher comes with a "Demo" Chat database installed, but you can add as many additional ones as you wish. Simply touch the "New" icon and then enter the name.

**Note:** A given slide show Chat can only have one Chat Database assigned to it, but each slide will point a different Chat Message Trail record in that database. However, you can have as many Chats as you want in your application, usually on a different page.

- c: Date and Time Stamping: Depending upon your selections, a date and/or time stamp will appear just above each Chat message in the Chat Trail.
- d: Visible Lines: A Chat trail can have up to 65,635 characters, so the full Chat trail could have many text lines defined. However, you can control the number of visible Lines that will appear on your phone at one time. This selection will be determined by the design choices that you make for your Chat. For most phones it is recommended that you have no more than 4 lines visible due to the screen space that the Photo and/or Video will take, as well as the Navigation Icons, if selected.
- e: Maximum Chat Width: Based on the choice, the Chat will be centered as best as possible relative to the Slide Show. There are 4 choices:
  - Slide Show Width
  - Twice the Slide Show Width
  - Three Times the Slide Show Width
  - Maximum Width
- f: Font Name, Style and Size: This sets the way the text will look, both in the Chat Trail and the Editable Text Field. Phones, however, usually only have a very limited number of Fonts and Font Sizes available. Express Mobile does its best to map your choices to what the phone can support. If you choose a Font Size of 9 or less, than for these limited Phones the "small" text size will be chosen. A size from 10 to 13 will map to the "medium" text size and over 13 will map to the "large" size.



### Publisher Interface

### Chat Operation on Wireless Devices

Chats adopt the characteristics of the device that they are being viewed on, so that the user experience is optimized.

The Chat above was designed and was running on a standard 3G phone, which usually has a screen size of 176 by 208 pixels, 2 soft keys, a joy stick, and 12 numeric buttons. Currently the Chat Trail for a particular application is being shown.



Fig 5-3

1: MIDP Form Behaviors:

Assuming that the default Text Entry Mode is MIDP Form, or the user has used the "Toggle MIDP Text Entry Form" to enable T9, then the behaviors, when that Text Field is selected, upon touching any numeric keys or FIRE are:

Initial State:



Fig 54

a: User touches the "space" key.

Creating Chat
Current Chat User: Jm
2008.03.26 (VVed) 08:47 PDT
User: MTP
2008.03.26 (Wed) 08:59 PDT
User: Adj
2008.03.26 (Wed) 08:59 PDT
User: Adjim
2008.03.26 (Wed) 09:00 PDT
User: Adjimmm
2008.03.26 (Wed) 09:08 PDT
User: Adwj
2008.03.26 (Wed) 09:14 PDT
Type in Chat

b: User enters "this is a test.". Touches Soft Key

Creating Chat				
Current Chat:59 PDT				
User: Adjjm				
2008.03.26 (Wed)	09:00 PDT			
User: Adjimmm				
2008.03.26 (Wed)	09:08 PDT			
User: Adw j				
2008.03.26 (Wed)	09:14 PDT			
User: Adww				
2008.03.26 (Wed)	12:32 PDT			
User: Hill this is a te	est			
2008.03.26 (Wed) Menu				
User: Hilthis is a t	Commit Edit			
	2 Alphe			
Type in Chat	3 ALPHA			
Hi this is a test	4 Number			
	5 Symbol			
	6 Mode			
	7 Cance			



2: Chat with Direct Text Entry Behaviors

Text will be entered directly into the message text field using the phones numeric key pad or keyboard.

a: Enter "Hi! This is a Test." Result

b: Touch FIRE Chat has been sent and Text Field Reset





Fig 5-10

The Soft Keys available are similar to that of MIDP Form text entry, except that there is no Commit command.



## B: Working with Phone Calls.

### 1: Manual Entry

Phone Number Text fields are automatically defined as numeric only data entry fields. They can be defined as:

• N. Am Phone #

Under the condition that 10 numeric characters have been entered, when leaving the Text Field the numbers will be formatted as:

(xxx) xxx-xxxx.

Other Phone #

No formatting is applied.

When either of these two types of fields are fired on that phone call is placed. On some phones it will be required to exit the application before the phone call can be placed, on others it will be placed immediately.

ľ	Setting	s E	vents 👘	Anin
	Define	the Tex	t Field St	ring
	J			
	Font Na	ame		Font
	Times	Roman	<b>~</b> ]	Reg
	Object	Name	·	Fran
	Text F	ield_1	-	Non
	Left	Тор	Width	H
	10	110	100	2
	Attribu	tes	Dra	w Frar
	Defaul	t	•	V
	Passw R <sub>Numer</sub>	ord ic		Supj
	N. Am.	Phone #		
	AN. AM.	SMS# Dhone#	≅ le	Dofo
٢	Other	SMS#	- E	Dela
)	URL Re	equest	¥۲ م	
	Audio I	JRL Req	UE 🛃	

Fig 5-12
2: Using Web Components

Various Web Components automatically generate Phone Lists.

Two examples are:

MapQuest Points-Of-Interest:

When using the MapQuest Points-Of-Interest feature, a list is returned with the names of those points-of-interest that were returned on the map. Firing on any location in that list will place a phone call to that location.

Face Book:

One feature for Face Book is to present a Buddy List Phone Directory. When firing on any name in that list a phone call will be placed to that person.

### 3: From Phone Contact List (PIM Data)

If the Phone List Message object is selected, and the phone supports access to the Phone's Contact List, then this list will automatically be populated with the

names on that list. Firing on any name will place a phone call to that person.

Multiple Chat
Video Chat
Phone List
Phone Field

Styles Help

**Ø**O

ents Pages

Fig 5-12

# C: Working with SMS Messages

#### 1: Manual Entry

SMS Number Text fields are automatically defined as numeric only data entry fields. They can be defined as:

• N. Am SMS #:

Under the condition that 10 or 11 numeric characters have been entered, when leaving the Text Field the numbers will be formatted as:

1-(xxx) xxx-xxxx.

In the event only 10 numeric characters were entered then the number "1" will be prepended.

Other SMS #:

No formatting is applied.

It is necessary to apply an Exit Event to this field "Send SMS". At that time the author must select either a Text Field or Text Area that will contain the SMS message. When either of these two types of fields are fired on that SMS message will be sent.



# **Define the Text Field String**

and the second	Font Name TimesRoman	-	Font Reg
	Object Name Text Field_1		Fran Non
and the second se	Left Top 10 110	Width 100	H 2
	Attributes Default	Dra	w Frar
R	Password Numeric N Am Phone #		Supj
Â	N. Am. SMS # Other Phone # Other SMS #		Defa
	URL Request Audio URL Requ		

Fig 5-12

2: From Phone Contact List (PIM Data)

When inserting this "SMS List" object into an application, the author must go to the Item Attributes Tab and select a Text Field or Text Area for linking that field's contents for the SMS message.

If the SMS List Message object is selected, and the phone supports access to the Phone's Contact List, then this list will automatically be populated with the names on that list. Firing on any name will send the SMS message to that person.

Eems	norn Annibuse String
Home	Call Home
ect Attribute	Use Rem Color
Line item Title -	r
	Use them Cost
www.tens.altratel	- I
and the second of the second	•
liam Fyants an	d Web Services
Frit Plater	
Place Phone Call	-
Send String on FIRE	-
Send String on FIRE or I	lumeric Keys
Goto a specific Page Vi	ew with Aleri
Goto Wildget	
Generate an Aleri on Fu	re di
Constant New York	
Toggle Alert with OnFor	us/Officus
Goto an Application with	h Aleri
uoto Start Page	*****
venerate as steri with	SAUCHA OUITION +
I' had a state of the second second	laningen til ing ing der til som
IC whind then an	na fankel
as#11010 Meisas	D PACIO



Fig 5-14

Chapter 05.11

Bindings

# D: Working with Rich Media Alerts.

1: Using Chat for Instant Messaging.

Section A of this chapter describes the various ways of chatting with another person that is connected to the Internet.

2: Using Web Components

One feature for the Face Book Web Component is to present a Buddy List. Firing on any name in that list will send an IM to that person's Face Book account. They will be notified of this pending message either through their Xpressmo Face Book Web Component Application and/or their Desktop Face Book Page.

Settings

Events

**Events and Web Services** 

Goto Widget

Send String on FIRE or Numeric Keys

Goto a specific Page View with Alert

3: Sending Rich Media Alert Messages.

The author can define Alert Pages for encapsulating rich media content as well as a user generated Instant Message. Included in this Alert Page could be dynamically defined Images or Video for sending real-time content from one person to others.

The following must be defined by the author:

An Alert Page:

This is where dynamic content such as Live Video, RSS Feeds, etc. could be linked.

A Message Field:

A Text Field or Text Area where the User generated message will be entered.

	<b>D</b>				<u> </u>	
Δ.	Pho	na	- NB II	nhor	- HO	id.
~	1 110	116	INU	INCI	110	<b>u</b> .

Generate an Alert on Fire Send SMS Message Toggle Alert with OnFocus/OffFocus Goto an Application with Alert Goto Start Page Generate an Alert with Synchronization Send SMS Message with Single Player Download Place Phone Call with Single Player Download = Send SMS Message with Multiple Player Download Place Phone Call with Multiple Player Download Send Instant Alert Message (IM) Set Logical First Page View **Popup Alert** Message Field My Message Pz 🔻 My Message

Animation

Color

Fig 5-15

This is where the user will enter the phone number. The author must place a "Send Instant Alert Message" Exit Event on the Phone Number text Field and select which Popup Alert and which Message Field they want to associate this operation with.

1 ğ] i Z, 0 Chapter 06 14 14 1 ie with 1.5 478 6 usiness Forms 

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 6: Working with Business Forms

A:	Introduction	06.04
B:	Text Fields	06.05
C:	Always On Text Fields	06.06
D:	Lists and Dropdowns	06.09
E:	Text Entry	06.23

Chapter 06.3

# Chapter 6: Glossary

List Layout Style	An extension of Object styles that apply to the default settings available for complex lists.
RSS List	One of many types of Lists that have available templates in the Pub- lisher that can be created and then subsequently be changed. The RSS List Template specifically defines a convenient way for assigning RSS feeds to a List and then displaying its fields in a user-friendly way.
Fire	The act of touching the FIRE key, touching the screen with s pointing device, clicking the mouse, or pressing the ENTER key on a keyboard.

Chapter 06.4

# A: Introduction

Business forms are a set of very powerful UI objects that can be used in many ways for, among other things:

- 1. Transactions
- 2. Sophisticated Navigation Objects
- 3. Enhancing Web Components.

All business forms can have the same set of Events, Colors, Animations and Bindings that are available to any other UI Object.

Text Fields, Dropdown Menus, and Selection Lists are further discussed later in this chapter.

Scrolling Text Areas behave as a browser-based Text Area would behave. The number of visible lines can be set, and a scroll bar will automatically be drawn if the number of text lines exceed that of the number of visible lines.

Check Boxes behave as a browser-based Check Box would behave.

Submit Buttons will, by default, send the current values for all the other business form objects up to the Express Mobile Transaction Server for processing.

Clear Buttons will reset the values of all the business forms on the page to their initial values when the page was first displayed.

Submit and Clear buttons have a "Don't Scale" state. If enabled the MIDP player will maintain the original width of the button object. This is useful if you are superimposing these type of buttons over rasterized text or images.

# B: Text Fields

The following types of text fields are supported:

- 1. Default (normal alpha numeric field)
- 2. Password
- 3. Numeric
- 4. North American Formatted Phone Number
- 5. North American Formatted SMS Number
- 6. Unformatted Phone Number
- 7. Unformatted SMS Number
- 8. URL Request (See Web Components)
- 9. Audio URL Request (See Web Components)
- 10. Purchase URL Request (See Web Components)

If Text Fields are of the types:

- 1: Numeric
- 2: Phone Number
- 3: SMS Number

Then the text entry model is always direct. Only numeric character can be entered. Cursor movement, editing and deleting within the string are also supported. When a direct entry text field is selected the "back" key, if available on the device, will similarly delete one character to the right of the insertion point.

In the event that the Text Field has the attribute Phone Number, then, under the condition that 10 numeric characters have been entered, when leaving the Text Field the numbers will be formatted as:

#### (XXX) XXX-XXXX.

In the event that the Text Field has the attribute SMS Number, then:

- 1: under the condition that 10 numeric characters have been entered, when leaving the Text Field the number "1" will be prepended .
- In either case, the number will be formatted as:
- 1-(xxx) xxx-xxxx.

When entering these text fields the formatting will be removed and just the numeric characters will display.

# C: Always On Text Fields

#### 1: Background

Certain applications may have pages where the text entry function should be always available, even as the user can navigate and operate on other objects on that same page.

For example, the page could have a Slide Show and a List. If while either is selected and the '6' key is touched, the focus and selection will switch immediately to the Always On Text Field and the letter 'M' will display and a blinking insertion point will appear. From that point on, the Always On Text Field operates as a normal data entry field, as specified in the Publisher.



### 2: UI Implementation

Using the example above the Application, with the Always On Text Field Selected is as shown below.



- There are no Text Areas in either the Master Page or the Application Page
- The Text Field is set to:
  - Selectable=false -
  - Editable=true

Chapter 06.8

3: MIDP Player Implementation

As shown in the Section 1, when the Text Field has been selected through a data entry event, the soft key 'Menu' will appear. Touching that Soft Key will offer the same choices as expected.



Any navigation event that is valid for a Text Field, such as an UP event, will cause the Selection Rectangle and Focus to move to the appropriate object on that page and the blinking insertion point to disappear, as shown in the graphic to the right.



# D: Lists and Dropdowns

1: Complex List Types.

Lists can be of the following types:

- a. Dropdown Choice
  - Default
  - Active Channel Personalization
  - Inactive Channel Personalization
  - Search From List
  - Widget Selection List from User's Library
  - Phone List
  - SMS List
  - · Widget Selection list from Runtime Widget Library
  - Phone List from PIM Data
  - SMS List from PIM Data
  - Search Response list
  - RSS List
  - Control List
  - Web Component List

#### b. Selection List

- Default
- Multi-Select
- Category List
- Channel Personalization
- Widget Selection List From User's Library
- Phone List
- SMS List
- Widget Selection list from Runtime Widget Library
- Phone List from PIM Data
- SMS List from PIM Data
- Search Response list
- RSS List
- Control List
- Web Component List

Services are supplied to support the following:

a. Personalization of the Channel Launch Strip

Dropdown lists have list types for both adding and removing channels (icons and their related intelligence) from the Channel Launch Strip.

If the Channel Launch Strip has the "Remember Channels" attribute selected, then in all cases the current state of the Active Channels will be reflected in these Lists, and the memory of the selected active channels will be saved across sessions.

b. Widget Selection from User's Library

The list object will automatically populate with the icons and the widget names that exist in the author's private widget library. No new items can be added or existing widgets deleted, but other attributes for each widget list item can be added.

c. Widget Selection from Runtime Widget Library

Both Lists have Widget Selection Lists that are automatically populated with an entry for each Widget that is available to that player. Each Widget, at authoring time, can be assigned a unique icon.

If an item in a Widget Selection list has a "Goto Widget Object" Exit Event then a Widget lcon Will be drawn. If the Widget Page has already had a Page lcon defined then it will be drawn. Otherwise the default Widget lcon will be drawn.



At runtime when firing on an Item in a Widget Selection List that Widget will automatically be loaded and executed on the device.

d. Xpressmo Server Page Selection

For both Lists that are set to their default type, if there are any of the following exit events associated with a List or Choice Item:

- Goto a specific Internal Web Page
- Goto the next Internal Web Page
- Goto a specific Internal Web Page with setting starting slide
- Goto a specific Internal Web Page with Alert
- Goto Logical First Page

and that internal WebPage has a Page Icon defined then it will be drawn. Otherwise, no icon will be drawn.

e. Automatic Population and Execution of Phone and SMS Lists:

Both Lists can be set to be automatically populated with the PIM data defined in the phone's contact list.

If the device supports and permits access to the PIM content list data, then when firing on a Phone List Item the following will occur. If the phone supports simultaneous use of the phone with an application then the Xpressmo application will continue, otherwise the user will be prompted whether they wish to exit the Xpressmo application and then place the phone call

If the device supports and permits access to the PIM content list data, then when firing on an SMS List Item the SMS message will immediately be sent to that contact.

#### 2. List Layout Styles.

All object styles have an associated List Layout Style. The author can create as many List Layout Styles as desired by Selecting the "Create List Button". If so, then that new List Layout Style will become the associated List layout Style for that Object Style. The author can always change which is the associated List layout Style for a given Object Style by changing the selected List object Style.



If the author wants to change the definition for a given List Layout Style then Select the "List Layout Style" in the Object style's Settings list will display the dialog box below.





The List Layout Style has the following controls:

- a. First Line:
  - Title Icon (optional)
  - Title String (mandatory)
  - Body Icon (Optional)
  - Body Text (Optional)

The body icon and text can be left or right justified.

The Title Text and body text can have a selection of either 2 different fonts, including color. The font size will always be derived by the Object Style Font Size for both fonts.

- b: Second line (Optional)
  - An indent can be defined relative to the first line.
  - All other settings available for the first line are available and optional for the second line.

- 3: Working with Selection Lists.
  - Selection Lists have "Settings" attributes that are defined at three distinct levels:
    - Object Level
    - List Level
    - List Item Level

You can switch between attributes levels by firing on the desired tab.

### **Object Attributes**

Object Attributes deal with the basic Selection list object, including the Object's font Style, number of List Items that are visible (which either directly or indirectly define the height of the List object).

Settings	nts Animation	Color
Object Attributes		Item Attributes
List Visible Items	Draw Frame	Object Style Default
Font Name TimesRoman 💌	Font Effects Regular 💌	Font Size
Object Name	Frame Style	Frame Width
Left Top 41 43	Width Height Lo	ck Hide
Remember Me	Suppress Selec	tion Rectangle
Add Persistent V	ariable	
I	Ľ	

Fig 6-10

### **Publisher Interface**

#### 4: List Attributes

List items can be added by Selecting the "Add Item" button. The initial Item name, and its corresponding "Title String" will be set to a default value of "ItemNumber-#" where # will be the number in the list that the new Item was placed in, and immediately appear in the List Items Dropdown menu.

The settings for Define List Item Layout are initially inherited from the List Layout Style that is assigned to the current Object Style when the Selection List Object was created.

Any other List Style can now be selected if desired by selecting a list Style from the List Style Dropdown menu.

Any changes made to the "List Item Layout" will override the inherited settings. Any changes in terms of the availability of Icons and/or Strings at the list item level will be reflected in the "Item Attributes" screen when a new Item is created. At the List Attributes level the following controls are available

Settings Events	Animation	olor 👝 🚺 Bi	ndings
Object Attributes	List Attributes	ltem Attr	ibutes 👔 🖬
List Items	List Styles	List Iter	n Layout
49ers win big! 💌	default 🚽	1st Line:	2nd Line:
Add Item	Leading	icon: 「	Г Г
Remove item	Line	Title: 🔽	Г
Remove All	Use Alt.	Font: 🦵	Г
	Body	icon:	. Г.
List Selector Type	Body S	tring:	۲.
Multiple Select	Use AR.	Font: <b>F</b>	Г
Personalize Channels Select Widnets from Us	Right Ju	stify: 🔽	Γ
Phone List	ernate Fort Nam	e: Alternate I	ont Style:
SMS List	Arial 🗸	Bold	<u> </u>
Phone List from PiM	ond Line Indent	Define Sec	cond Color
Unselected Item Icon			
None	<u>م</u>		
			Fig 6-11

1. For Selection Lists of types Default, Search Response, RSS Lists, Multiple Select, Category, Select Channels and Select Widgets, the following elements can be defined..

#### a. First Line:

- i. Title Icon (optional)
- ii. Title String (mandatory)
- iii. Body Icon (Optional)
- iv. Body Text (Optional)
- v. The body icon and text can be left or right justified.
- vi. The Title Text and body text can have a selection of either 2 different fonts and/or colors.
- b. Second line (Optional)
  - i. An indent can be defined relative to the first line.
  - ii. All other settings available for the first line are available and optional for the second line.

- 2. For Multiple Select Lists a palette of both selection and unselected vectors and icons are available.
- 3. For Category Lists a palette of vectors and icons are available.
- 4. For Search Response Lists:

The author can create a fully defined Search Response List form the Text Dropdown Menu.

Four objects will be created:

- Search Response List
- Text Field
- Text Area
- Slide Show

They will be logically linked together. The author will then be presented with three dropdown lists for possible defining the logical relationship with any Search Response List Item and other related objects on the page that could display metadata that is associated with that response. The maximum number of possible Search Response List Items can be set.



The Three Logically Linked Objects are:

• Title Object

This could be either a text button or a text field, the difference being whether the width is fixed with possible scrolling or the width adapts itself to the length of the title string.

If a text field, then the "Title Marquee Enabled" check box becomes enabled. If selected, then, if the string for the title is too long for the width of the text field, a marquee animation will begin when this search result is selected.

Message Object

This could be either a paragraph or a text area, the difference being whether the number of visible lines is fixed with possible scrolling or the height adapts itself to the number of lines after the text string has been formatted to the width of the text area

Media Object

This must be a Slide Show. Any image or video metadata will be attached to slide that is in the same relative position as the list item.

Settings Events	Aŋ
Object Attributes	List
List Items	Li: di
Add Item	
Remove Item	
Remove All	
List Selector Type Search Response	J
Objects for Title None	
Objects for Message Text Area_2	Alte [#
Objects for Media Slide Show_3	Sec
Title Marquee Enabled	
Max Number of Respon	Ses
Failed Search Messa	qe ]
	F(= C 1 2

During runtime, the object names for the Title, Message and Media object will be used to dynamically update their content as each item in the Response List is visited. As with all list items, any exit event can be assigned.

The author can also override the default error alert that will be displayed by the player if the search returns no responses.

<u> </u>	
No responses	detected.
	OK Cancel Hel

### **Publisher Interface**

5: For RSS Display Lists:

The author can create an RSS Display List in a manner very similar to that of a Search Response list.

In the Template available to the right, The Title and Message (abstract) Objects are defined and logically connected to the RSS Display Lists.

This template also places, if available, the Channel's Image lcon as the title icon. The lcon is automatically sized so that its height will be equal to the Item's height, and the width is then calculated to maintain the image's aspect ratio.

This is the second example of a Web Component utilizing the same technology as that of a Search Response List. See Chapter 11 for a full description.



Fig 6-15

The author adds channels by Selecting the "Add Item" button under the "List Attributes" tab of the Resource Inspector. In addition to selecting the RSS Channel, the author can set the maximum number of feeds that will be extracted and displayed on the mobile device.

The author can remove any of the channels or remove all of the channels as desired. The Title Object String is dynamically bound to the 1st line title string of List. The Abstract Object String is dynamically bound to the 2nd line body string of the List.



### Publisher Interface

#### Item Attributes

The graphic to the right shows the choices if a String item element is selected. In that case there are choices on whether to apply the Alternate Item Color and/or the Alternate Item font. Any choices will override the defaults that were inherited from the List Layout. The "Item Attribute String" can be edited and will immediately appear in the Workspace's Selected List Object. If the "Selected Attribute" is the 1st Line Item Title then this will become the List Item name as displayed in the "List Item" Dropdown menu under the List Attribute Tab.

	and a second state of a second state of the second state of the second state of the second state of the second	and a second	
Settings Stevents An	imation 2+ Co	or 👔 🖉 Bindin	js 🖌
Object Attributes	Actributes	Item Attribute	s [
Select Attribute	ttem Attribu	te String	
1st Line Item Title	49ers win big!		
C	, Ilen i	tern Color	
Remove Item Attribute	0381 F		
	,	tom Fout	
	USEI		
	1		
Item Events a	and Web Service:	5	
Exit Player			
Place Phone Call			
Add Icon			
Remove Icon		······································	
Goto a specific Page	view with Alert		
Goto Widget			
Generate an Alert on I	-ire	8	
Seria SMS Message			
Goto an Annlication w	ith Alert	لقسا	
Goto Start Page	azə e 4964 t		
Generate an Alert with	n Synchronization		
Kalina Statement Statements	- (III		
		Damum Alant	
		норир жиеп	
default 🔽		SystemAlert	<u> </u>
			Fig 6-17

The graphic to the right shows the choices if an Icon element is selected. If "Select an Image Icon" is touched then the "Icon Management" dialog box is displayed.

Note that the 1st Line Title attribute can never be removed.

Settings Events A	nimation Co	lor Bindings	
Object Attributes	st Attributes	Item Attributes	
Select Attribute	Selected Icc	on File	
1st Line Leading Icon 🛛 🔻	file:PublisherAc	consininers.png	
Remove Item Attribute	Select an Im	iage Icon	

Fig 6-18

#### 5: Working with Dropdown Choice Lists.

Touching the "Add Item" button will display a dialog box where the item's attributes can be fully defined. If these attributes are to be changed, then select that item in the "Drop-Down Items" Dropdown menu, and Select the "Edit Item" button.



Fig 6-19

The differences in List Selector Type from a Selection list are as follows:

- Multiple Select and Category Selector Types are not available.
- Personalization of the Channel Launch Strip. The "Select Channels" selector type is replaced by "Active Channels" and "Inactive Channels".
- Search From List

The "Search From List" Object can be populated in the same ways as other lists.

When unselected, the "Search From List" Object will draw in a manner similar to that of a Text Field. When selected the following actions will occur:

For the indirect Text Entry Mode:

- On pressing Fire, or a text or numeric key the page view will switch as in a manner similar to that of a Text Field with the Text Field on the top of the screen representing the last successful search, if any. The search list items, starting from the first valid match (in alphabetical order) will be drawn below the Text Field.
- Characters can be deleted and/or entered.
- As soon as the first character is entered the Selector List will open.
- If the entered character is valid (the current search string has at least one partial match to a search list item) then:

- If there is only one unique match then:
  - · The page view will return to the original page view
  - The unique search list item will be selected.
- If there is more than one list item partial match then the search list items drawn below the Text Field will be redrawn, starting now from the first valid partial match of the current search string..
- If the character entered defines a string that does not match any search list items, then that illegal character will be removed.
- There will be two soft key commands.
  - Cancel:

The page view will return to the original page view with no change.

Commit

ĉ

The following behaviors will be supported:

- If the string entered is completely valid and matches, at least in part, a minimum of two items in the list, then:
  - The first item that has a successful partial match will be selected.
  - The dropdown will be open for navigation to other possible search list items.
- The search list items in the dropdown will start from the currently selected item.
- · Search is not case sensitive.

6: Working with Selection List Libraries.

If a dropdown list or selection list is selected and a "Save As" operation is requested, then there will be a new Choice under File Types.

Save As File Menu		<u>. X.</u>
File Name		
Listname	Generate OVGA Form Facto	17:1
File Type	Generate QCIF Form Facto	r: <b>Г</b>
Application	Embed Applicatio	n: <b>Г</b>
Current Web Page	Embod imagool(Ciido	e- <b>Г</b>
Widget Object	chibed intrages shae	3,1,
Application Template	Embed Audio/Vide	o: <b>Г</b>
Style 1 emplate	Recal Mah Commonat	e• 🗖
Selected Object Style	react from an process	-v≁r
	, 	
		<u></u> _
	OK   Cancel	IHell

These Saved Lists, including all their icons and dynamic binding, can now be imported into an application, either from a public Xpressmo library or from the User's List Library. This process begins by touching "Import List" under the "File" Menu.

Selection List Library	
Private Selection List Library States desited	Alabama
	OK   Cancel   Helg

### Publisher Interface

#### 7: Icon management.

Icons are similar to images in that they have both private and public libraries and can be seamlessly uploaded from your PC to your private library.



#### 8: Suppress Frame

Only applies to List Objects.

If this checkbox is Selected and the number of items in the list are less than or equal to the number that are visible, then the Frame drawing will be suppressed.

#### 9: Non-scaling Lists

By choosing the "Define Form Objects by Pixels" under the "Project Menu" the List will now be defined similar to that of an Image. The location and size are specifically defined, and they can be manipulated by dragging an attachment point.

The font size, when in "Define Form Objects by Pixels" mode, will be calculated based on the number of items that are visible and the height of the List object.



Fig 6-23

## 10: Working with the Edit Dialog Box.

You can work with a Selection List utilizing the Edit Dialog Box which is available from the Layer Inspector or by right clicking over the Selection list Object.

	List item	5	Visib	le Nems	Lines	per (Second/Tour	ch)
bject Definition olor Settings fouse Events htty Animation	ESPN Font Nan TimesRu	er nan v	6 Font Bold	▼] Effects		0 • Font Size	Add Rem Edit Rem Remove Item
ain Animation ait Animation Aiki Objects	Object N List Obje	ame ct_2	Generic Ο Fram Nom	bject Attri a Style	Frame Width	Draw Frame	Remove All List Selector Type Select Channels
Restore Default	LOCK	Hide M	Left 10	Top 10	Width 200	Height	Feedback Margin       2     -
		ESPN Pattots b SF 49ers Davidson se Netflix Special p	E 44,9 Seat Colta 35-14 E 52,9 locted as coach Prome rome for today.		Selected It Nane Selected S	em Icon	Define item Layou Unselected item ic None
							OK Cancel H

If the "Define Item Layout" button is touched then, in a manner identical to the "List Attributes" screen in the resource inspector, the Item Layout settings, which were inherited from the Object Style, can be changed.

	Leading icon	Line Title	Use Alt. font	Body Icon	Body String	Use Alt, font	Right Justify
st Line: 🌾	▼ 「	2	Г	P	4	S I	9
nd Line 🖇	7 F	ম	Г	Г	R	P	4
leivetica ont Style Sold Italic	- -	Ist Line	Title			× Ist Line	Body
econd Lin	e Indent	2nd Line	Title			2nd Line	Body
efine Sec	ond Color						

When touching either the "Add Item" or "Edit Item" buttons, the dialog box will have an "Extended Item Attributes" button. This will display the "Extended Item Attributes" dialog box which will work in a manner similar to that of the "Item Attributes" screen in the resource inspector.

# E:Text Entry

#### 1: Background

For languages that can be sufficiently supported using Unicode characters 33 through 126 (this excludes, among others, Chinese, Japanese and Korean) it is possible to support direct text entry as long as the device returns to Java the keycode that was touched. Currently all phones with numeric key pads will be supported. MIDP 2.0 devices with soft keyboards currently will have to continue to use the indirect "MIDP Form" text entry model. Also, all CDC and J2SE devices will also be supported.

For MIDP 2.0 devices with soft keyboards we probably will need to rely on the Response Director to inform the player that it must use the "MIDP Form" text entry model. The current implementation will assume a numeric keypad so it will not work on MIDP 2.0 devices with soft keyboards.

### 2: Supported Objects for Direct Text Entry

The following objects will be supported in build 2.1.46:

- Text Fields
- Chat Messages
- Selection Lists
  - Used for Forms to minimize touches
  - · Used for prompted Search to assist User in selecting a useful search term

.

## 3: UI

In addition to all the other attributes that can be assigned to Text Fields it is possible for the author to set the default text entry mode.

The user can subsequently change the input mode (See "4: Behaviors" on the following page).

Settings	Eve	ents	Animatio	on (	olor	Bindings
Define th	e Text	Field Str	ing		Obje	ct Style
ſ	******			-	Default	
Font Nam	18		Font Effe	cts	Font	Size
TimesRo	man	Ŀ	Regular	-	16	•
Object Na	ame		Frame S	tyle	Fram	e Width
User nan	ne		None	•	1	-
Left 1	Гор 110	Width	Heigh	t Loc	k Hide F	
Attribute Default	s	Draw 되	Frame	Sele F	ctable 7	Editable
Datatoli Passwor Numeric	ď		Suppres	s Select	ion Recta	ngle
N. Am. Pl N. Am. Sl Other Ph Other Sk	none # MS # one # 1S #		Default 1	Text Entr	y is MIDP	Form
		J				Fie

- 4: Behaviors
  - a: Direct Text Entry Behaviors:

When a supported object is selected it will appear as a text field with a blinking cursor.





Test	
	Fig 5-27



A text insertion point will be active and will blink 2.5 times per second.

There are seven possible text entry states for Standard Text Fields.

Exit Fig 629

5 Mode 6 Clear

There are eight possible text entry states for Selection Lists and Chat Text Fields which can be selected through soft key commands.

Celifornia	Menu
Colorado	1 Clear All
Connecticut	2 Clear
elaware	3 Open Launchstrip (*)
	4 Alpha
	5 Number
	6 Symbol
	7 ALPHA
	8 Mode
	9 Exit
122.25	N
*olitensisettettettettettettettette	Fig

	Menu
2008.03.26 (Wed) 12	Clear All
2008.03.26 (Wed) 12	Clear
User: Hi! this is a te <mark>3</mark>	Open Launchstrip (*)
HB 4	Alpha
5	Number
6	Symbol
7	ALPHA
8	Mode
9	Exit
Exit	Menu
	Fig 6-31

- Alpha
- In this state touching a numeric key will cause an immediate echo of the first character assigned to that key. As long as the delay before touching that same key is less than 1.6 seconds the insertion point will not move and the next character assigned to that key will be echoed. If the number of touches exceed the number of characters assigned to that key the choices will be repeated. If either a different key is touched or the same key is touched after a 1.6 second delay, then the insertion point moves one character to the right, previous character is shown, and the new character, as appropriate, is echoed at the insertion point.

Capitalization will occur if the character is an alpha character and it's the first character in the string, or the character is immediately preceded by a space character.

The current implementation is identical to that of the Sprint implementation for Samsung A900 phones. (See appendix A)

ALPHA

Same behaviors as with "Alpha" except that all characters entered will be in uppercase.

Numbers

In this state touching a numeric key will cause an immediate echo of the numeric character assigned to that key and the insertion point will move one character to the right.

Symbols

When in this state a palette of symbols will be drawn on the top half of the screen, with a selection rectangle drawn either around the symbol in the upper left, or the last symbol selected. Navigation will work in both dimensions, with recycling when hitting a symbol palette boundary. Once the desired symbol is selected FIRE will place the symbol in the text field at the insertion point and the symbol palette will be erased. A soft key command is available to escape from the symbol palette. See Appendix A for list of symbols available from the Symbol palette.

Clear

Clear will delete the character to the left of the insertion point but will only be available when not on the leftmost character.

Clear All

All the content in the text field will be removed

Mode

The user can switch between Direct Text Entry and MIDP Form text entry. This gives the user access to T9 support.

The following additional soft key Command will be available for Chat Messages and Selection Lists:

"Cancel"

In both cases the field will be cleared of all characters.

The following Navigation Keys will be supported.

LEFT

This will move the insertion point left one character. If already at the left most character then this will cause the focus to leave this object and the next appropriate object will be selected.

• RIGHT

This will move the insertion point right one character. If already at the right most character then this will cause the focus to leave this object.

FIRE

For Chat Messages and Selection lists FIRE will complete the operation. For example:

- With Chat the chat message will be sent to the user's audience as well as become visible, with the defined time stamp and user handle, preceding the message.
- For Selection Lists that are inputs to a Search Function, the partial or complete search term will be sent to the search engine for responses.
- For Selection Lists used for minimizing touches when selecting a string from a list, the Selection List will close and display the current value as defined below.
- For Text fields the Exit Event, if any, will be executed.
- BACK

If available, it will delete the character to the left of the insertion point. However, when not on the leftmost character, the "Back One View" operation will be executed.

UP and DOWN

These will remain as object selection keys when on a text field. If on a Chat Message or List then their meaning has already been in the Chat and Complex List chapters of the User's Guide.

Special Behaviors for Selection Lists when in Direct Text Entry Mode

When unselected, the Selection List Object will draw in a manner similar to that of a Text Field. When selected the following actions will occur:

- As soon as the first character is entered the Selector List will open.
- · Characters can be deleted and/or inserted.
- If the entered character is valid (the current search string has at least one partial match to a search list item) then:
  - If there is only one unique match then the unique search list item will be selected and the Selection List will close.
  - If there is more than one list item partial match then the search list items drawn below the Text Field will be redrawn, starting now from the first valid partial match of the current search string.
- If the character entered defines a string that does not match any search list items the character will be inserted but the selection list items will not be redrawn.
- While the selection list is open the UP and DOWN operations will be supported.
- If FIRE is touched:
  - If there is a first valid partial match with the selected item, or the item selected after navigating via the UP and DOWN operations, that item is selected and the list will close.
  - If there is no match then the list will close with the current non-matching entry will remain visible.

	Fig 6-3	2
d		, i
California		
Colorado		ŝ,
Connecticut		
Delaware		

Fig	6	3	
-----	---	---	--

	 Address of the Address of the Addres		
d	21	N) - 1 	୍
California	 		=1
Colorado			
Connecticut			
Delaware			

Fig 6-34

b. MIDP Form Behaviors:

Assuming that the default Text Entry Mode is MIDP Form, or the user has used the "Toggle MIDP Text Entry Form" to enable T9, then the behaviors, when that Text Field is selected, upon touching any numeric keys or FIRE are:

**Chat Text Fields** 

Initial State:



Fig 6-35

**Touches Soft Key** 

Creating Chat	
Current Chat:59	PDT
User: Adjim	
2008.03.26 (Wed)	09:00 PDT
User: Adjimmm	
2008.03.26 (Wed)	09:08 PDT
User: Adw j	
2008.03.26 (Wed)	09:14 PDT
User: Adww	
2008.03.26 (Wed)	12:32 PDT
User: Hil this is a te	est.
2008.03.26 (Wed)	Menu
User: Hil this is a t	1 Commit Edit
r	2 Alpha
Type in Chat	3 ALPHA
Hithis is a test	4 Number
	5 Symbol
	6 Mode
	7 Cancel
	Fig 6-37

d: Touches FIRE

Chat has been sent and Text Field Reset



Fig 6-39

xpressmo.com

# Creating Chat Current Chat User: Jm 2008.03.26 (VVed) 08:47 PDT User: MTP 2008.03.26 (VVed) 08:59 PDT

a: User touches the "space" key.

b: User enters "this is a test.".

User: Adj 2008.03.26 (VVed) 08:59 PDT User: Adjim 2008.03.26 (VVed) 09:00 PDT User: Adjimmm 2008.03.26 (VVed) 09:08 PDT User: Adw j 2008.03.26 (VVed) 09:14 PDT

Type in Chat Hil

c: Touches "Commit Edit" Result



Fig 6-38

Chapter 06.30

Selection Lists

$\square$	
	Fig 6-40

Initial State

When the first character is entered a Form displays in which the search string is on top, and below it are the search candidates, sorted in alphabetical order, that are equal or higher in the collating sequence of the search String.

Searching List.	and the second
Type in Search String	
d	
Search Candidates	
California	
Colorado	
Connecticut	
Delaware	
Florida	
Georgia	
Hawai	
Idaho	
lünois	
Indiana	
lowa	
Kansas	
	Fig 641

As with Direct Text Entry, characters can be deleted and/or inserted.

All other Direct Text Entry behaviors, with the exception of navigating with UP and DOWN events through the search candidate list, are also supported.

When a unique search string (as shown below/left) Occurs, then the Form will close and the final result will appear as shown below/right.

Searching List	California
Type in Search String	Contornio
Са	Fig 6-43
Search Candidates	-
California	
Colorado	
Connecticut	
Delaware	
Florida	
Georgia	
Hawaii	
idaho	
Minois	
Indiana	
iowa	
Kansas	
Fig 642	

#### Standard Text Fields





	·		
# EXPRESS MOBILE User's Guide

Chapter 07 Working with Media Objects

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 7: Working with Media Objects

A:	Pictures (Bitmap Images)	07.04
B:	How to Edit Using Photo Enhance	07.05
C:	Slide Shows	07.09
D:	Video	07.16
E:	LBS Maps	07.19

Publisher Interface

Chapter 07.3

# Chapter 7: Glossary

Library

All media objects can have both private and public libraries, where frequently used media content can be placed for subsequent use in future editing sessions and with any application.

Slide Show

.

A very powerful collection of Images and or Videos, organized in various ways, visualized in various ways, and behave in various ways. The author has full control over these options.

# A: Pictures (Bitmap Images)

Pictures may be imported from:

- The designer's local computer
- The designer's private library
- · The public library

Any picture that is imported from the designer's local computer will, when the application is saved, automatically uploaded and placed into the designer's private library for subsequent use with other applications and/or pages in the current application.

If the picture is resized, a scaled variant of that picture will automatically be created, and placed with the original in the application's Image folder. For performance and bandwidth efficiency, it is the scaled version of the picture that will be downloaded to the network connected device when the application is being executed. This automatic scaling also occurs for all pictures that have been placed in Slide Shows.

If the picture was in the GIF format, then when the application is saved a PNG version of the picture is automatically created. This means that for phones that only support PNG there will be valid picture for downloading. This process also happens automatically.

# B: How to Edit using Photo Enhance

The "Photo Enhance" feature allows you to add many different effects to your images.



#### 1. Image Filters:

Use this feature to add different filtering effects. Some examples are below.



xpressmo.com

Fig 7-2

### 2. Crop:

This will eliminate unwanted portions of your image and enlarge the area selected. Click on any of the drag points with your mouse and drag until you reach the desired size. Note that the "Crop" icon has changed into "Apply Crop". Click "Apply Crop".





Area selected for Crop

After Crop

### 3. Text Message:

Add a text message to your image by clicking this icon. A box will appear in the image; click in the box and type your message.

By selecting the dropdown boxes, you can change the font type, size and color. You can place the message anywhere on the image by dragging while selected. You can also add a frame around the text message by choosing from Frame Style and establish its width by choosing from Frame Width.



Fig 7-10

4. Rotate: 💽 Rotate your image by increments of 90°.



 $\mathbf{X}$ 5. Delete:

Used to delete text that was added to an image. Make sure the text area is selected first.

6. Reset:

To return the image to its original state, click the "Reset" icon. Any and all changes you made will be eliminated.

7. Typeface, Font Size, Color:

Change the font type, size and colors by using these dropdown menus.

Typeface	Font Size	Color
Helvetica Plain 💌	6 pts 🔀	Black 🗶
		Fig 7-13

8. Frame:

Choose a frame style and frame width for the text message by using these dropdown boxes.

Frame Style	Frame Width
None 💽	0 pixel 💌
None	· P
Raised	
Pressed	
Solid	
	Fig 7-14

9. Name and Compression:

Shows the file name of the selected image and how much it is compressed (similar to reducing resolution in other photo edit programs).

> Name: Desert m~.jpg Compression: 70 Fig 7-15

#### 10. Bright, Sharpness, Color

Original

Original

These three sliders allow you to change the brightness, contrast (sharpness) and color of your image. You can also type a number in directly.

Brigm	Sharpen	Color
50	50	50
T.		( <b>*</b> **
1 1	• •	Fig 7-16



Brightness:

Sharpness:

Fig 7-21 Fig 7-22 Fig 7-23

80

30

Color:



· 30

# C: Slide Shows

Slide shows are very powerful objects, which can include photos, audio and video. When extended to a Chat Session (see "Working with Chat" in the "Working with Messages" Chapter) a slide show can become a real time multimedia community communication facility.

#### How to Define a Slide Show

For a simple slide show there are a number of attributes under the "Settings" tab that are unique.

• Max. Visible Slides:

The number of slides that would be visible at the same time. If the total number of slides is less than this number then only the actual number will be drawn.

Vertical:

Whether the Slide show should be considered to be in a vertical or horizontal alignment. This will affect the following:

- How Multiple Visible Slides would be drawn.
- How the navigation keys would work.
- Where the Scroll Bar would be drawn (if any)
- Chat Room Status: (See Messaging)
- Initial Slide:

This will be the slide that will be first drawn when the page is initially entered. The default is the first slide.

 Add Persistent Variable (See Persistent Variables)

Settings Events	Animation	Color Bindings
Max. Visible Slides Ex	kplode Effect	Object Style Default
Vertical Explode (%)	Scroll Bar Width	Pixel Separator
Object Name PPT_ss	Frame Style	Frame Width
Left Top Width 0 1 240	Height Loc 259 F	ck Hide Dim Page
Select Scroll Bar Positi Below Slide	on Worl	k With Slides
Horizontal Icon Strip Se Bottom: Pans Up	Ting Vertic	cal Icon Strip Setting ert to Slide Show 💌
Horizontal Active Icons	Vertie	al Active Icons
Chat Room Status	Fram	e Delay(milliseconds)
Chat Room Settings	Numl	er of Frames
Chat View Only	Reme	s Selection Rectangle
Initial Slide Slide1	Add Pe	rsistent Variable

**Object Selected: Slide Show** 

Fig 7-27

Work with Slides

This choice opens up the Slide show Dialog Box as shown below.



### 1: Import: 🖉

By clicking this icon, you can import a video clip, single image, a folder of images or an existing image album.

NOTE: the maximum image size for a slide show to be used on a computer is 380 (h) by 440 (w) pixels. For phones and PDAs, the maximum image size will be defined by your screen dimensions

Import Video:

.

Select (mage(s), Video of Slide, Libraries	Select Video by URL
Import Image or Video	video Library
Import image Folder	ANSTRUMENTS AOL_Radio_AK_Fallin.5gp AOL_Radio_NJ_FAV.3gp AOL_Radio_NJ_FAV.3gp Being_Webixed=_2.3gp
Image Library	Personal Video Library
Enter Slide Library	AK_Fellio_64.3gp
Video Options	
Cancel Help	
Fig 7-29	OK Cancel Hel

Import Video Clip

Express Mobile Video Library

To add a personal clip, touch "Import Image or Video" button. This allows you to choose which clip to add from your files. Once the clip is selected, touch the "Slide Settings" button and under "Delay", select "Stay" or a time sufficient for the Video to complete. To play, select the "Play" control.

Import Photographs:

- a. To import a photo from the Express Mobile library, touch the "Image Library" button.
- b. To import a photo from your private library, touch the "Private Image Library" button.
- c. To import a single photo touch "Import Image or Video" button.
- d. To import a folder of photos touch "Import Image Folder", double click on any of the photos in the folder or touch "Open". All of the photos in that folder will be imported.

Import Slide Show Library Objects:

In addition to the above, there is a library in which you can save or access finished slide shows. To get to the library, touch "Enter Slide Library". This will bring you to the library and you can then choose which show you want to import, either from your own private library or from the Express Mobile Library.



Fig 7-31

# Sice second

#### 2: Slide Settings:

This feature lets you define, animate and link your images.

a. Animate

This feature allows you to change the timing and look of an individual photo. Note that you can have different attributes for some of the photos in your gallery, while others can be the same. This gives you some interesting options for how your photos will look when played.

- 1. Slide Pause Time: this is the time each photo is on the screen.
- 2. Animated Transition: determines how the next photo will appear on the screen. "Zoom In" and "Zoom Out" create a 2 times zoom of 1 slide to the next with the zoom centered at the center of the slide. These options work well if you have a series of images (such as GIS images) that zoom into an area or object. By using the slideshow, it will appear that there is an almost seamless zoom in or out. "Expand Over" zooms in once to the center of the image. "Shrink Out" zooms out once from the center if the image. Random is a compilation of all the animations.
- 3. Transition Time: the amount of time it takes to transition from one photo to the next.
- 4. Frames/Second: how smoothly the photo transitions. This is a function of how many frames of the photo display in 1 second. The more frames per second means a smoother transition.
- b. Replace Media

To replace an existing video clip or photo touch the "Replace Media" button.

c: Select an Exit Mode for this Slide:

By selecting this feature, you can link a video clip or slide to the next slide, an external web page or a specific slide. When the slide show is playing, a new web page will open when the show reaches a video clip or slide that's been linked. You can have the video clip or slide automatically go to the Web page, or have it be activated by a mouse click by selecting the "Mouse Enabled" checkbox.

If the video clip or slide is exiting to an external Web page or external page in a new window, you must enter the URL of that site. To go to a specific slide, select the number.

To go to a specific internal Page Number the Mouse Enabled function must be selected. With this choice the slide will time out after the Pause Time has elapsed and the next slide will be shown unless the user selects the slide while it is in its paused state. See Chapter 14 for more details.

d: Bubble Help

Bubble help can be defined, so that when that slide is defined this string will appear, centered directly above the Selected Slide.

e: Fire Enabled

If selected, visiting a slide through a normal navigation event (up, down, left or right) will execute the exit event of the newly visited lside.

### 3: Enhance Photo:

This takes you to the Express Mobile Photo tool where you can add special effects to a photo, such as adding

Enhance Photo The Control of Create Collage	Red Eye Removal	Crop Rot	کې ate D	elete		Essage Clip Ar	Pos	Card Order Pi	ints He	
Image Filters	File Name	Frame	Style	Frame V	Vidth	Typelace		Font Size	Colo	IT.
Inactive	demo_Slide01_a	None	-	0 pixels	-	Times Bold	<u> </u>	12 pts 💌	Black	-
Weave								Bright S	harnen	Color
Etch								50	150	50
Point								100	100	100
Sphere					۶Yf	1055000			and l	
Gamma		4444			· ^}	16931110				100
Threshhold		·							Care Care	
Gray		i Ri	ch &	Intera	acti	ve		1.54		200
Emboss		*						100	25	
Solarize		Yo man	20m	NO.31		×				
Diffusion		1.4.8	9	1 MP CO	DIN					
Dither		105-2				->T		(da		
Marble					( • • •	26.5				
Distant				~ 5		and the second s		62	444	
Oval			Deliv	erina th	e PC				23.3 2507	
		ex	nerie	nce on i	hon	es				
			P - 11			00		199		
			1	These -	a	-		<b>₩</b>		100031
		(				E.			أستنب	ليني
								ок	Can	el
in the second		-	24		an a	55 A <b>L 196</b> A	·			ining the second second



filters, text messages, and cropping. After you've made the changes, click "OK" and you'll be returned to the Slide show Dialog Box. For detailed information, see Pictures in the "Working With Media Objects" Chapter.



You can add an audio track or audio stream to your photo and/or the transition between photos. You can also have a continuous background track with the ability to superimpose a separate track for an individual photo or transition.

- a. To add a track for the time that the photo is on the screen, either type in the file pathname of your own clip or select from the audio library and click "open". You can have the track play once, in a continuous loop or stream.
- b. To add a track to the transition between photos or the gallery background, follow this same procedure.



Fig 7-33

5: Delete: 👗

This deletes the currently selected photo.

6: Colors 🧏



Fig 7-34

### 7: Generic Settings:

Choose how you want your photos to play and transition. These choices determine how the entire gallery or album will look and play. To change an individual photo only, select the "Slide Settings" button.

Fire Enabled:

If selected it makes all Slides mouse sensitive, and when touched, will execute the defined exit event associated with that slide, if any.

Slide Show Transition:

Determines how each photo will enter the screen Random is a compilation of all the transitions.

- Delay:
- · How much time the photo is on the screen.
- Transition Time:
- How long the transition takes from one photo to the next.
- FPS (Frames per Second):
- How many frames of a photo play per second. The higher the number, the smoother the transition will look.
- Scroll Bar Position:

The scroll bar can be hidden or placed in 6 different positions relative to the Slide Show. If the Slide Show has been set as Vertical, then the Scroll Bar will draw vertically and its placement we be relative to its horizontal position.



Fig 7-35

#### 8: Numbering: 1 of 16

This shows you the currently selected image in numerical order.

9: Controls:

Similar to controls on a VCR.

From Left to Right:

- a. Reverse: go back one frame at a time
- b. Pause: stops the slide show at the current image
- c. Play: begins or resumes the slide show
- d. Forward: advances one frame at a time

10: Image Index: To go to a specific image, click on it in the index.



You can change the order of the images by clicking and dragging. In the example below, the first image is being moved to the second place. The current #2 image will become #1. You can move any image to any location.



Moving #1 to #2



Result

### D: Video

The author has a number of choices for the source of the video:

- 1. Import from Client Computer allows author to select a video file from the file system on the Publisher client machine. Selecting this will open a file browser window.
- Video Library allows selection of video files on the Publishing server at Publishingserver/Libraries/VideoLib. This folder's content is available to all Publisher accounts. The content of this folder is determined by the System Administrator responsible for Publisher.
- 3. Personal Video Library allows selection of video files on the Publisher server at Publishingserver/~PubAccount/ VideoLib, where "PubAccount" is the Publisher account name. This folder's content is available only to the "PubAccount" Publisher account. Whenever a video file is imported from the client computer, a copy of the file is placed in this folder, so that it can be used for subsequent references either within the same application or from a separate application.
- 4. Import From a Page Library:
  - a. Page
  - b. Alert
  - c. Widget
- 5. Import From a Slide show Library

6. From the Libraries accessible with the Open Command:

- a. Style Template
- b. Application Template
- c. Application

The first three options were, and still are, available when adding a video object to a page, or when adding a Video Slide to a SlideShow object.

The only option that remains file based is "Import from Client Computer". In all other cases in the Publishing Client and in the Xpressmo Players the Video is URL based and will be streamed from our Streaming Server during any of the following Operations.

- Publishing Client: (Video, from an "Opened" application, will be streamed if the user is not operating in "LocalHost" mode)
- Play inside the Slide Show Dialog Box.
- Preview
- Generic Player
- MIDP Player

#### Select Video By Url

The "Create Video Object" dialog will be modified so that a 4th option is available, that of selecting a video using a URL. This is the first time that our platform will be streaming rich media content from any streaming server; no longer limited to the streaming server that is an extension of our platform.

Create Video Object
Select a Live Video Stream
Import from Client Computer
Video Library
AK Falin_64.3qp
Personal Video Library
OK Cancel [Help]
Fig 7-39

115 / 3

When "Select Video by URL" button is selected, the dialog box shown below is displayed:

A Shire Video 6/181	ij
Entervideo URL (must start with http://, https://, or rtsp://)	and the second second

The text field should be sufficiently wide to accommodate a reasonable length URL. If this dialog was visited before, then the last video URL entered will be present, so that the author can edit it, on the assumption that the URL's are alike. (In the future, there will be a dropdown from which to select the 10 most recently used URL's.) Pressing "Cancel" will cancel the specification of a video source by URL and return to the previous state, either the "Create Video Object" dialog or the Resource Inspector "Settings" tab.

Pressing "OK" will initiate a validity check on the URL. The URL must start with "http://", "https://", or "rtsp://". If not, a popup will display indicating the error. When the popup is dismissed, it will return to the "Select Video by URL" dialog so that the author may correct the URL.

	2002C4/2000
ising or has an invalid scheme	٦
ttp://, https://, and rtsp://.	accession of
<u> </u>	
Fig	7-41
nng hei bain UPL-avcaded	œ,
string has been converted;	1.00
	<u>ΓΟΚ</u> Fig mathematic states and the states of the states

To; [rtsp //cxamplo.com/%7Edamo]

The URL is automatically converted to a safe URL string. If the URL is already URL-encoded, it will honor the byte encoding and prevent double encoding. If the author's string is modified in any way, a popup will inform the author of the change. This popup is informational only. Once the popup is dismissed, it will return to the Publisher main layout

Fig 7-42

#### **Publisher Interface**

An attempt to connect to the resource pointed to by this URL is made. If connection cannot be established a Warning popup will appear, displaying the error condition reported by QuickTime. There may be valid reasons why a resource is not available at authoring time (such as if the resource has not been created yet) so this is popup is only cautionary. Once the popup is dismissed, the UI will return to the main layout of Publisher.

e Warning UTL	recource not accessible	
The URL'	s resource was not acce	essible.
	<u>. •04.</u>	, , , , , , , , , , , , , , , , , , ,
harmon and the second se		

Once validated, the Publisher will be set to store the value from this field in ObjectString[1] in the Video Object's parameters within the PDL file, when the application is saved. Similarly, the VideoObject's ObjectIntegerAttribute[2] is set to (-1). This indicates that this video object's source is from a URL rather than from a file that originated on the Publisher client file system or from one of the Publisher server "libraries".

The UI will return to the main layout of Publisher, showing a placeholder for the video on the layout page. The size of the Video will be initially determined by the size of the encoded video stream. The author may edit the size of this video object using the Resource Inspector or by manipulating the object directly on the page layout.

The video URL field is not editable in this tab. Any change must be made through a dialog box that will be displayed when button labeled "Change Video" is selected. This will display the same dialog as "Select Video by URL" dialog, renamed "Change Video by URL", with the current URL value in the text field, if a Select Video by URL was previously used. Otherwise the Create Video Object dialog will appear.

The Video, as with all videos, will be viewable in "Preview".

Video Mode	Change	Video Strean	nl	Object S	style
With Audio Mute Audio			וי	Default	<u> </u>
	Pathnam	e for Video S	tream		
4/xpressmo/	xpresme.c	om/~o/WebS	ites/Foo	1/VideoLib	ifoo4.m
				anna treadh tre a	
Object Name		Frame Style	9	frame V	Vidth
Object Name Video Object		Frame Style	<u>9</u> <b>→</b> ]	frame V	Vidth
Object Name Video Object	_1	Frame Style	•	Frame V	Vidth
Object Name Video Object Left Top	_1 Width	Frame Style None Height	∎ - Lock	Frame V 1 _	Vidth
Dbject Name Video Object Left Top 10 10	Uldth	Frame Style None Height	₽ ▼ Lock	Frame V 1 Hide	Vidth
Object Name Video Object Left Top 10 10	Uidth 100	Frame Style None Height 100	∎ ▼ Lock	Frame V 1 Hide	Viđth
Dbject Name Video Object Left Top 10 10 Stream Video	Uidth 100 Video	Frame Style None Height 100 Settings	Lock	Frame V 1 Hide F	Viđth

#### Video in Slide Shows

A slide show could always have multiple video slides, but they never worked very well in terms of:

- Streaming
- Slide Transitions
- Player Support

Now Slide Show based video is at the same level of support as Video Objects. In the MIDP Player there remains a restriction that only one video can play at a time, but in the generic player multiple video slides could be visible at the same time.

## E: LBS MAP



#### 1: UI

Selecting the LMS Map Icon from the Tool Bar will generate a fully functional LBS Map which is another example of a Web Component. This LBS Map object will inherently support panning through the Left, Top, Right and Down navigation keys. Zooming will have a zoom-in default but that can be changed through auto-generated Soft Key Commands or automatically when the map has reached its maximum zoom-in or zoom-out resolution.

A scroll bar with an elevator is drawn on the left to indicate the current zoom level.

In addition, hidden text fields are generated and initialized to default values, which can be changed by the author. They include:

- width: Set to the width of the view port plus160 pixels, which, by default is the width of the current page plus160 pixels less the width of the scroll bar.
- height: Set to the height of the view port plus160 pixels, which, by default is the height of the current page plus 160 pixels..
- street: An optional input that, by default, is set to the street address for Express Mobile.
- zip: A required input field that, by default, is set to the zip code for Express mobile.
- Lat: This is the geospacial latitude for the current location. It will either be set by the phone if it has a supported GPS receiver, or by the Web Component when the address is transmitted.
- Lng: This is the geospacial longitude for the current location. It will either be set by the phone if it has a supported GPS receiver, or by the Web Component when the address is transmitted.
- mapScaleforURL: This is the default zoom level, which, by default, is at a resolution of 48000:1.

All of these hidden text fields can be edited by the author, and all have been assigned persistent variables so that other pages in the application can either define or receive these values.

Chapter 11 will go into greater detail in how Web components can be used for a wide range of LBS applications.

#### 2: How it Works:

The all Map Images that are initially loaded are centered relative to the view port, one image map for each zoom level. If a pan operation is selected, then the existing image map will pan until it reaches the boundary of the view port. Simultaneously, the Web Component Manager requests for a new set of map images, whose geospacial center will conform to the geospacial point after the 80 pixel offset is reached.

An image map is brought down to the device for each zoom level, utilizing Xpressmo's multi-threaded Network Communication Manager, with a priority given to the image map for the current zoom level.

This means that, in general, there is no perceived delay by the user when panning or zooming.

xpressmo.com

IPR2021-01146 Page 00345

dê G 8 Chapter 8 ×. 1.5 Working with (\* † (\* † Vector Shapes \$ ¥. 3

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 8: Working with Vector Shapes

A:	How to Create	08.04
B:	How to Create a Group	08.06
C:	How to Edit	08.07

# Chapter 8: Glossary

The act of selecting two or more UI objects and then logically con-Group necting them together so that a single operation can be applied to all these objects at the same time. Arc One of several vector graphic primitives that can be created with the Publisher. Arcs are circle segments that can be used for creating curved surfaces. Fill The color that will be drawn inside a vector object. If set to transparent then no color will be drawn. Gradient A technique for creating the sense of depth for a two dimensional vector object. This is down by using two different colors in various ways to create effects such as sunlight reflecting off a 3 dimensional object.

Chapter 8.4

# A: How to Create:

There are three convenient ways to create vector shapes.

- Touch the "Shapes" icon from the Toolbar. The menu below on the left will appear.
- Select "Shapes..." from the Object Menu from the Menubar. The menu below on the right will appear
- Basic Shapes Arrows Lines More Shapes Library... Fig 81

Draw Tools

• Right Click on the Canvas with no object selected and select "Shapes...". The menu below on the right will appear



In all three cases you can also access the vector shapes libraries, which contains a number of shapes already created, either by the designer in the private library or by others in the public library. Click on your choice and the shape will appear on the Web page with a red selection frame surrounding it, with the exception of Free Polygon, Polyline or Ellipse drawing tools. You can now edit the object.

To create a freeform polygon or polyline, click on the first icon as above and a crosshair will display. To begin, click where you want the first part of the shape to begin. You can then drag the mouse to design the shape. Alternatively you can click the mouse wherever you want the next point to be defined. At any time, touching and releasing the shift key will cause the next line segment to be horizontal or vertical. Hold and move your mouse to define the shape desired.

To draw a straight line other than horizontal or vertical, move the mouse (without clicking) to the point where you want the line to end and then double click the left or right click your mouse.

Publisher Interface

To end the drawing, either double click (left) or right click your mouse. The shape is now ready for editing.

A freeform polyline is similar to a freeform polygon, except the lines don't need to form a closed-in object. Follow the instructions as for polygons to create



To create an ellipse, click on Ellipse drawing tool. A message will display prompting you to create the arc (a. below). Click where you want the center point of the arc to be and click again where you want the other side of the arc to be. A message prompting you to define the arc will display (b. below). Draw where you want the arc to be (c. below). You can use the drag points on the frame to make the arc into the shape desired. (d) and (e) below show edited versions of the arc.



The starting point of the angle and/or the angle sweep can be changed by editing (see Section C).

# B: How to Create a Group:

If you have a group of shapes that you want to stay together for placement purposes, you can choose to group them. To do so, hold down the "Shift" key and then with the mouse, choose the objects to be grouped (a red selection frame will be around each object).



To group, touch the "Group" icon in the Commands or right click and select "Group".

The group will have a boundary box around it and will be listed as a group in the Object Status Box as in the example below. Groups are designated with the letter "G".

If you wish to group all objects on the page, you can go to "Edit" dropdown list and click on "Select All" or right click the page and click "Select All" at the bottom of the list.

To ungroup, click the "Ungroup" icon or select "Ungroup" from the Edit menu or do a double left mouse click and select "Ungroup".

The group can be sized by dragging any of the red drag points.

Polygon groups can be saved in the Polygon library for future use. Make sure the group is selected and click "Save As"; choose "Selected Shape Group" and name it.



# C: How to Edit:

To access the Edit mode, make sure the shape is selected and either:

- Select the Edit Icon on the Layer Inspector.
- Select "Edit" on the Edit menu from the menu bar.
- Right click and select "Edit"...
- Work with the Resource inspector.

#### 1: Define:

Editing Shapes Other Than Arcs:

Although all editing operations are available with the Resource Inspector the discussion below assumes that editing is occurring though the Edit dialog box.



Fig 8-12

Select a Point: each shape is made up of a number of points, which are represented by the blue squares. To choose a particular point, you can click in the dropdown list and select. The point selected turns into a red square and then change the value in its horizontal or vertical position. The "Move Polygon Point" button will become active.

٠

in the angle degree desired.

The Example below shows the result of changing the Vertical Position to 40 and Selecting the "Move Polygon Point" button.

	Select a l	Point	Line Viedth	Rotate	Angle	Horizoatal	
Chiest Definition	Contract		1	10	Construction of the local distance of the lo	139	
Color Settings	Point Inde	al 🗐	More Potype	an Point		Vertical	
Ercats	Point Inde	n 2 🖙	Ontoto Pobra	An Dane		40	
Lacry Abimation	Point Ind	11.5 10.1	MANAN CHIM	<u> </u>		,	
Lait Animation	1		Generic Object Attributes				
Child Objects	Object Na	ume	Frai	ne Style		Frame Width	
	Polygon	Objec1_2	Not	e v		1 •	
Suppress Selection	LOCK	Hide	Left	Top	Width	Height	
Г	r	r.	89	40	101	101	

Fig 8-13

• Line Width: changes the width of the shape lines. The following example has a line width of 10.

Rotate Angle: rotate a shape by a specific selected angle. Type

Fig 8-14



xpressmo.com

# IPR2021-01146 Page 00353

Note that you can also do a free form rotate by selecting the shape and doing a double left mouse click or a single right click. A message saying "Rotate Mode" appears; begin to move the object with the mouse to the desired rotate angle.



- Object Name: the default name is "Polygon Object". Change the name to something more descriptive by typing in a new name.
- Frame Style: this adds a frame around the shape. The choices are raised, pressed or solid.
- · Frame Width: determines how wide the frame around the shape is.
- Lock: prevents the shape from being dragged or resized by using the mouse. Other editing functions are still available.
- Left/Top: this changes the placement of the shape on the web page.
- Width/Height: changes the size of the shape.

Editing Ellipses:

	Rotate Ar	ngle	Line Width	Start Ang	le	Angle Sweep	
Doject Definitions Color Settings vents ntry Animation	0		10	62		-256	
lain Animation			Generic Ol	oject Attributes			
Child Objects	Object Name Frame Style			e Style	Frame Width		
	Arc Obje	ct_2	None	<u>ا</u>		1 -	
Suppress Selection	Lock F	Hide F	73	Тор 26	Midth 58	Height 149	· .
•*-, • <b>2</b> ,-12-,-	•		$\bigcirc$	- 20° 91			

• Line Width: changes the width of the shape lines. The following examples show an initial line width of 1 and then of 20.



Line Width of 1

Line Width of 20

- Rotate Angle: with an arc you can also rotate the shape by a selected angle. In addition, the start angle and angle sweep can be adjusted.
- You can perform a free form rotate by right clicking your mouse and selecting "Rotate" from the menu. A message saying "Rotate Mode" appears and when you click, drag the mouse to define the desired effect.



Below are different examples of how the start angle and angle sweep look at various settings for ellipses.



- Object Name: the default name is "Arc Object". Change the name to something more descriptive by typing in a new name.
- Frame Style: this adds a frame around the arc. The choices are raised, pressed or solid.
- Frame Width: determines how wide the frame around the arc is.
- Lock: prevents the arc from being dragged or resized by using the mouse. Other editing functions are still available.
- Left/Top: this changes the placement of the arc on the web page.
- Width/Height: changes the size of the arc.

- 2: Color and Fill:
  - a. For Arcs (not polygons):



Fig 8-24

Color Target allows you to set the line, fill and frame color. You can use either the color palette or create a custom color by using the red, green and blue dropdown boxes. The feedback box gives you a preview of what your choices will look like.

b. For Polygons:

You have the choice of a Single Color fill, a Linear Gradient fill or a Radial fill.

i. Single Color Fill:

See above for ellipses.

ii. Linear Gradient Fill:

This divides the shape object into 4 quadrants. You have a choice of 2 fill colors. The percentage of each color can be adjusted by changing the gradient mid point.

The origin of the color changes, depending on the gradient angle.

#### Chapter 8.13

#### Publisher Interface



Fig 8-25 Gradient midpoint of 50 and O'Gradient Angle

Figure 8-25 shows the editing of the colors for a polygon using the Edit dialog Box. Working this way, the gradient angle arrow can be dragged or adjusted by changing the value in the Gradient Angle text field.

In Figure 8-25 above, the 1st color fill is red and the 2nd color is blue. The mid point is set at 50, which means there is an even amount of the 2 colors and the gradient angle is at 0.

Gradient midpoints change the percentage of the 1st fill color. Utilizing the Resource Inspector, with the Color Tab selected, the same functions can be performed. As seen below, at 20, the percentage of the red (1st fill color) is decreased.



The result at 70% is shown below.



Fig 8-27 Gradient midpoint of 70 and 0°Gradient Angle



Below are the results when the gradient angle is changed to 80, 180 and 270 degrees.





Blue: 255

Transparent Γ

+





Fig 8-30 Gradient midpoint of 50 and 270°Gradient Angle

#### iii. Radial Fill:

Radial Fill has a center point from where the color radiates out. Using the same colors as for the Linear Gradient, Figure 8-30 below shows the fill radiating from the center of the object and a gradient midpoint of 50. The percentage of the 1st fill color increases or decreases when this number is changed





Fig 8-30 Gradient Midpoint of 50

The X & Y midpoints can also be changed, so that the center point changes location. The X coordinate can be from 173 to 369 and the Y from 52 to 231.
· · ·

xpressmo.com

## IPR2021-01146 Page 00361

EXPRESS MOBILE User's Guide

Chapter 09

Working with Text Objects

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 9: Working with Text Objects

 A:
 How to Create
 09.04

 B:
 How to Edit
 09.05

Chapter 09.3

# Chapter 9: Glossary

Lines per (Sec/Touch)

This is the vertical scrolling speed for a multiline text object such as Text Areas and Paragraphs.

## A: How to Create



#### 1: Button:

Click on either the "Button" lcon (a) or "Text" and then "Button" under the "Objects" menu (b) or right click and choose "Button". The button will be placed onto the Web page with the words "Button Text" and a red selection rectangle around the button; type in the desired text. Editing is now enabled, as are many commands and icons on the status lines. To move the button to another location, click on the button text (but don't release the mouse button) and drag to the desired location. Note you can move the button before entering text if you wish.

2: Paragraph:

Click on the "Paragraph" Icon (a) or "Text" and then "Paragraph" under the "Objects" menu (b) or right click and choose "Paragraph". The paragraph will be placed onto the Web page with the words "Paragraph Text" and a red selection rectangle around the paragraph; type in the desired text. Editing is now enabled, as are many commands and icons on the status lines. The procedure for moving the paragraph is the same as for text above.

3: Variables:

Click on "Variables" under the "Objects" menu (c) above. Your choices are listed in the graphic to the right. Audio Mute Control will mute any audio track that may be present. The other choices will be placed on the screen and you can then edit and drag as for buttons and paragraphs.



Chapter 09.5

## B: How to Edit:

The Edit function works the same for buttons and paragraphs with the exception of the "Define" tab. Editing a variable is identical to editing a button, except the text cannot be changed.

	]	Define the Te	xt Button St	ri <b>na</b>	***************************************		ar an - c <b>a</b> an
<b>Object Definition</b>	Expres	s Mobile					
Color Settings	Font Na	me	Fo	nt Effects	•	Font Size	
Entry Animation	TimesR	loman 👻	B	old Italic 💌			
Main Animation Exit Animation	<b></b>		Generic	Object Attrib	utes	P	
Child Objects	Button	vame Object_2		one 🔫			
Suppress Selection	Lock F	Hide F	Left 40	Top 140	Width 130	Height 27	* *
		Expr	ess Mob	ile			
		Dofino for	Buttons				Fig &5

Define for Buttons:

"Edit the Text Button" allows you to edit the text that was entered when the button was first created.

🖾 Edit Object Attribu	tes		• • • •				art. Cotos
Max ObjectDescriptions Color Settings Events Entry Animation Main Animation Exit Animation Child Objects	Gimum Width 50 Font Nam TimesRom Object Na Paragrap	e man v nobject 4	dth %) S N F( R Generic Fi	crollable one v ont Effects egular v c Object Attribu rame Style	Line utes	s per {Second/To 0 - Font Size 16 - Frame Width	ouch)
Suppress Selection	Lock	Hide T	Left 74	Тор [40	Width 95	Height J66	
		This i Expri Paraj	is an ess Mobili graph	C			Fir 85

Define for Paragraphs

Define for Paragraphs:

"Maximum Paragraph Width as a % of Screen Width" lets you decide how wide a paragraph should be in relation to the screen width.

"Scroll" and "Lines per (Sec/Touch)" apply to paragraphs that have been bound to a real-time text feed such as RSS (See appendix D).

The remainder of the "Define" tab is identical for all. You can change the font, font effects (regular, bold, italic or bold italic), and font size (from 4 points to 5000 points). The object name can be replaced by a name that is more descriptive, such as "Button1". The frame style choices are: none, raised, pressed and solid.

If you have chosen a frame to outline your text object, you can select the frame width from 1 to 200. The frames above are set at 8.

"Lock" allows you to lock the object in to prevent it from being dragged or resized by using the mouse. All other editing functions are available.

The "Left" and "Top" selections are for the position of the button, paragraph or variable in number of pixels from the top or left of the screen.

2: Color:



Fig 8-6

Color Target allows you to set the color of the text, button or paragraph background and frame. You can use either the color palette or create a custom color by using the red, green and blue dropdown boxes. There is a feedback box; in this example the text color Foreground color) is red, text background (background color) is yellow and frame (frame color) is blue.

# EXPRESS MOBILE User's Guide

Chapter 10

Working with Child Objects

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 10: Working with Child Objects

A:	How to Create	10.04
B:	How to Select	10.06
C:	Behaviors	10.07

Chapter 10.3

## Chapter 10: Glossary

Behaviors

Child Objects

Describes the intelligence of an Object as it responds to User interaction, time, or location.

Used for creating dynamic visual effects or progressive disclosure of information

A child object may be:

- Vector
- Video
- Image
- Slide show
- Paragraph
- Text button

There are no limits to the number of child objects that can be attached to any particular parent object. Similarly, there is no limit to the number of parent objects for any given Web page. Therefore, this can be an extremely powerful feature.

There are 3 general functions:

- 1. Selection: when a parent object is selected, all child objects appear.
- 2. Fire: the child object is not visible until the parent object is fired on. To enable this feature, see Section D later in this chapter.
- 3. Timelines: a) a regular timeline that is activated at the time the parent object is activated and b) a timeline event attached to a child object. See Chapters 13 and 14 for a detailed description of timelines.

## A: How to Create:

To create a child object, make sure the parent object is selected.

Then either:

- · go to the Menu Bar, click on "Events" and then on "Child Objects".
- Go to the Events Tab of the Resource inspector and touch the "Work with Child Objects and Mouse Overs" button.
- · right-click on the selected parent object and select "Child Object".

#### Chapter 10.5



A "Child Object Mode" floating palette will appear. The author can now, without limit, place child objects on the page from any of the choices as shown on the right. As each child object is placed on the page, it becomes selected as shown below. The author can change from one selected child object to another through mouse selection. Each child object can then be further defined as described below.

Selecting the "Close" button will exit the Child Object mode

**Publisher Interface** 



## B: How to Select:

When Child Objects are created they will appear in the Layer Inspector as seen above. Selecting a Child Object from the Layer Inspector will:

- 1. Cause an implicit switch into Child Mode.
- That Child Object will be selected as shown above in green while its associated parent object will be selected in red.
- All the relevant controls in the Resource inspector that were available to a Parent Object will be available to a Child Object.
- 4. The Layer commands are available
- 5. The Delete command is available
- 6. Undo is available.
- 7. The child object name can be edited in the resource inspector.
- 8. All other child objects associated with the selected parent can be selected at this time for editing.
- 9. To implicitly leave child mode:
  - a. Touch any parent object in the Layer Inspector.
  - b. Touch any Parent Object in the Workspace.
  - c. Deselect all objects by either touching a free space in the workspace or by selecting "deselect all" from "Edit" off the Menu Bar.



## C: Behaviors

a: Visibility

There are three possible techniques for causing child objects to change their visibility.

· Selection of the Parent Object

Upon selection all child objects will appear (subject to their timeline definitions, if any) and will disappear when the parent object is deselected.

Toggle Children on FIRE

When this attribute is selected for the parent object then selection or deselection itself will have no effect. The first FIRE on the parent object the child objects will become visible and remain that way until another FIRE event occurs on that parent object, or when another Parent Object is FIRED on that has the "Hide non-related Children" attribute selected.

b: Child Slide Shows, Timelines and Animations



Child objects can be assigned all the timeline and animation attributes that a parent object can be assigned, and

a Child Slide Show can be assigned all the slide transition animations that a Parent Slide Show <sup>Fig 106</sup> can be assigned.

The activation of these visual and audio effects begin when the child object becomes visible and ends whenever the child object loses visibility or the timeline is completed, whichever occurs first.

.

# EXPRESS MOBILE User's Guide

Chapter 11

Working with Web Components

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 11: Working with Web Components

A:	Supported Objects	11.05
B:	How it Works	11.07
C:	Behaviors and UI	11.10
D:	Adding a Web Component	11.11
E:	Completing a Web Component with the "Generate UI" Option	11.13
F:	Completing a Web Component with the Existing UI Objects	11.19
G:	Persistent Variables	11.23
H:	Enhancing Web Components with Auto-Generated Persistent Variables	11.25
l:	Restrictions for Editing UI Objects related to Web Components	11.29
J:	Working with Control Lists	11.31
K:	Working with Collection Lists	11.38
L:	Collection List Keys and Complex List Indexes	11.40
M:	Sharing a Web Component	11.42
N:	How to create a Web Component Model	11.47

•

Chapter 11: Glossary	
Web Component Widgets	Unlike the widgets a discussed in Chapter 2, Web Component Widgets are a series of one or more related pages that employ Web Components. Often a Web component Widget will execute a Web component immediately upon being activated.
Data Types	Data types represent a specification for each input (Input Data Type) to a Web Service or output (Output Data Type) from a Web Service. A data type will describe the format of the data that the Web Service provides and how that data may be organized. Data Types become helpful hints so that:
	<ul> <li>The Publisher can make an intelligent decision on what type of UI Object to generate when auto-generating.</li> </ul>
	<ul> <li>Present the author with a concise list of possible and valid UI Object can- didates with creating a Web Component manually or when Editing a Web Component and changing the binding from one UI Object to another.</li> </ul>
Symbolic Field Name:	This is the unique symbolic pointer to a specific input or output that is pro- vided by a Web Service.
Binding:	Binding is the mechanism which ties a Symbolic Field Name as known to the Web Service to a UI Object as known to the Publisher through its Object Name.
Control List:	A Control List is a an Output Data Type that has the following characteristics:
	It has a related Symbolic Field Name that points to a Web Service Output.
	Its Symbolic Field Name is prepended with "Control List" for easy identifica- tion and is drawn in red.
	It aggregates a set of Output Symbolic Fields that that are logically related to the Control List's Output Symbolic Field. In data structure terms the Control List's Symbolic Field is an index to a data record that contains logically related data fields, each of which has an Output Symbolic Field reference.
	Some examples of Control List Implementations are:
	1. Search Response List Control List
	The short Title for each of the returned matches could be the Control List Symbolic Field.
	Related Symbolic Fields:
	a A detailed description
	b. A related image banner Ad

c. The UKL to the web site would represent rela	ited.
---	-------

2. RSS Display List

The Title for each RSS Feed could be the Control List Symbolic Field.

Related Symbolic Fields:

- a. The Summary
- b. A related icon image

c. The URL to the full story web page ..

3. Store Front Product List

The name of the product could be the Control List Symbolic Field.

Related Symbolic Fields:

- a. Product Description
- b. Product Preview (Image, Video, specification,, Audio, etc.)
- c. The URL to the WAP Store Front

Child Output Objects:

Collection Lists:

The UI Objects that have been bound to a set of Output Symbolic Fields that that are logically related to the Control List's Output Symbolic Field.

Collection Lists are Lists that contain a set of hidden attributes that point to a set of Control List definitions. In data structure terms the Collection List's set of hidden attributes are keys to a table of Control List Definitions.

## A: Supported Objects

The following objects will be supported in this first implementation:

- 1: Input UI Objects:
  - Text Fields
    - Alpha
    - Numeric
    - Phone Number
    - SMS Number
  - Text Areas
  - Choice Objects
    - Returning the Selected Visible String
    - Returning a Numeric Hidden Attribute.
  - Single Item Selection Lists
    - Returning the Selected Visible String
    - Returning a Numeric Hidden Attribute
  - Multi Item Selection Lists
    - Returning all selected Items
      - Visible Text String
      - Hidden Attribute
    - Cluster Item Selection Lists
      - Returning the hidden attributes for all items
  - Check Boxes
  - · Slide Show
    - Returning a Numeric Hidden Attribute
    - Returning a String Hidden Attribute
    - Returning the hidden attributes for all slides
  - Submit Function (Can be assigned to any object including Submit Buttons, Vectors, etc.)

2: Response UI Objects:

1.Single Line Text Object

- a. Text Field
  - i. URL
  - ii. Audio URL
  - iii. Purchase URL
  - iv. Other
- b. Text Button
- c. Submit Button
- d. Clear Button
- 2. Multiple Line Text Object
  - a. Text Area
  - b. Paragraph
- 3. Check Box
- 4 .lmage
- 5. Video

6. Slide Show (with either Video or Image slides, or both)

- 7. Choice Objects
- 8. List objects

9. Control Lists (they control all the subordinate Output UI objects for that Web Component.

- a. List Type:
  - i. Search Response List
  - ii. RSS Display List
  - iii. Other
- b. Choice Type:
  - i. Search Response List
  - ii. RSS Display List
  - iii. Other

Chapter 11.7

## B: How It Works

Both Xpressmo customers and third parties can register their web components directly into the Xpressmo Authoring Platform.

1: Xpressmo XML Web Component Registry

Customers and third parties can use XML to register their Web Components into the Xpressmo XML Web Component Registry. The Registry may contain references to:

- All Consumer Inputs related to the Web Component.
- Any Environmental data such as PIM, time or location values.
- Any persistent variable data.
- All outputs related to the Web Component.

Appendix A shows the ten current Web Components currently registered into the Xpressmo Web Component Registry.

- RSS: USA Today Top Stories
- Yahoo Search
- Google Search
- MapQuest
- Yahoo Maps
- Yahoo Weather
- Accuweather
- Movies
- ThumbPlay Store Front
- Send SMS Message
- Stock Quotes
- Face Book



2: Binding to the Author's Application

The XML Web Component Registry for all registered Web Components is loaded into the Publisher Client tool at login time. The author can then assign any of these Web Components to an application without any need to write code. Each Input and/or Output related to the Web Component may be hinted by the Web Component Registry Administrator. The more hinting the more information is passed to the Application Designer in terms of presenting a better focused set of possible UI Object candidates for binding. As a Web Component is selected the authoring platform presents the author with dialog boxes that enable the binding of all the inputs and outputs of the Web Component to GUI components of the application. Multiple Web Components can be assigned to any UI Object or Event in order to facilitate mashups. These UI Object and/or Event bindings, for each instance of a Web Component, are stored in the PDL. The war file containing a WebComponentManager to handle data from player to web component automatically is deployed as a web application archive to any application server.

Again, looking at Appendix A, some of the key Fields are:

• Parameter Name:

This is the Symbolic Field Name that is displayed to the Author.

• DataType:

This is where the Web Component Registry Administrator can add hinting to help focus which types of UI Objects can be selected by the designer.

ParmType:

If defined, it is set to "Request" in order to inform the Web Component Manager how to process this information.

Use:

This can be set either to "Required" or "Optional". If set to Required, then the Author will be required to assign a UI object to this Field, and the Consumer will be required to have data entered if the Required UI Object is an Input. In either case, warnings will be issued if the Required Fields are not handled correctly.

#### ServiceMapping:

This is the Symbolic Field Name that is sent to the Web Component Manager for communicating with the Web Component. It uses XPath to map the output result to the result XML Element or Attribute.

Comment:

This is information that the author will see displayed in the Comments field for each Symbolic Field, to assist the author in understanding the meaning of the Web Component.

- In the event that the Web Component returns lists of results then there is an additional Field named Control-Parm:
- This is the Symbolic Field Name that will point to an Output UI Control List object. When used, the Control
  List manages the visible presentation for all other Output UI Objects associated with a particular implementation for a Web Component in a given application.
- 3: Execution of Web Components

The Xpressmo Player, upon detecting an event in which a Web Component has been defined, assembles and sends all related inputs to the WebComponentManager URL which proxies the request to the 3rd party, customer, or Xpressmo web service and returns outputs to the Xpressmo Player. The Xpressmo Player then takes the outputs and binds the data to the UI components in the application.

For example, if the author wanted to use an ATOM feed in their application, they would look at the UI Components available in the Publisher Client, pick the feed they want to use, and bind the output of the feed summary to a textbox. The bindings would be saved into the PDL and processed by the player at runtime. If the ATOM feed does not exist a new one can be added to the Registry that contains all the configuration data required, such as the actual feed URL, the WebComponentManager URL, and what output fields are available for binding.

## C: Behaviors and UI

Under the Bindings Tab there are new choices related to binding Web components with any page or any object on a page. The drop down will list all those Web Components that have already been bound to this Object. These Web Component choices are available at all times, regardless of whether an object is selected or not.

When one or more Web Components have been bound then the "Edit" and "Remove" buttons will become active.

Dynamic Binding is still available, and can be used simultaneously with any Web Component.

Settings Events . Ani	mation Color Bindings
Web Component and V	Veb Service Operations
	Add Edit Remove
Attributes Exposed	
Page Height Page Height Background Image Background Color Page Delay Time Transition Animation Transition Time Frames per Second Page Audio Track Transition Audio Track Page Exit	<ul> <li>         ∰ Player         <ul> <li></li></ul></li></ul>
in a second to the second s	
Default Attribute Val	ue Link Set
Database Name T	able Name Field Name
Channel Name C	hannel Feed Operation
Reuters 🚽	1 Replace
bject Selected: Page	
	Fig 1

## D: Adding a Web Component

When initially touching the "Add" button the screen to the below appears, with the first Web component that has been registered selected. In this case it is the Web component that still relies on our Feed Collector for presenting any of dozens of RSS feeds to the application. In this case, since this is a Web Component implementation of the RSS Display object, "Generate UI Objects" will be selected.

- Generate UI Objects, if selected, will automatically generate the UI Objects as described below. If not selected, then the author will bind the Web Component Inputs and Results to previously created UI Objects.
- Select Results Page is used when you want the Results page to be a different page then that of the Input page. The default selected results page will either be set to the current page, or, if there are both inputs and outputs, it will be set provisionally to the next page in the current page order, if one exists.
- Add Web Component Select a Web Component Select Results Page USS Display List 🐁 🔹 97 S. . Map -MapQuest-Directions.getDirections MapQuest-GetMapCluster-Pan.panN Activation Options MapQuest-GetMapCluster-Pan.panS Preload -MapQuest-GetMapCluster-Pan.panE MapQuest-GetMapCluster-Pan.panW Generate Ul Objects MapQuest-GetMapCluster-Pan.getMapCluster V MapQuest-GetMapSingle.getMapSingle MapQuest-POl.getPOI Share Web Component Pictures Г SendSMS Weather2 Weather Movies Stocks FBActions.getFriends FBActions.setStatus FBActions.sendIM FBActions.get(M FBActions.uploadPhoto Logon WebSearch RSS1 AliTop10.getAliTop10 RingTone-Top10 SearchAll Search Search-Ringtones Mapsimage OK Cancel Help
- Activation Options include, if there are no

Input UI Objects, a choice to either "Preload" the Web Component (similar to how Dynamic Binding currently works), or have the Web Component executed when the "Results" page is viewed by the consumer.

- Share Web Component becomes available and will become selected under the following conditions:
  - A Web Component is Selected which already has been used by the current application.
  - The current Input page is also a "Result" page for that Web component.

This permits the consumer, after viewing the results, to extend the Web Component (See Section 4) allow the consumer to make additional queries against the same Web Component. Some examples are interactive panning and zooming for a Mapping application, or additional and or refined searches for a Search application.

xpressmo.com

Fig 11-3

The example below shows how "Generate UI Objects" will be selected for a Mapping application.

MapQuest-GetMapCluster-Pan.panN MapQuest-GetMapCluster-Pan.panN MapQuest-GetMapCluster-Pan.panS MapQuest-GetMapCluster-Pan.panW MapQuest-GetMapCluster-Pan.getMapCluster <u>KDPAUESUCCUMAPSINELSCOLUSTORS</u> MapQuest-POI.getPOI Pictures SendSMS Weather2 Weather2 Weather3 Movies Stocks FBActions.getFriends FBActions.getFriends FBActions.getFriends FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 SearchAll Search Search Ringtones	Select a Web Component		r	Selec	t Res	ults	Page	а Л
MapQuest-GetMapCluster-Pan.panN MapQuest-GetMapCluster-Pan.panN MapQuest-GetMapCluster-Pan.panN MapQuest-GetMapCluster-Pan.panW MapQuest-GetMapCluster-Pan.getMapCluster <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster-Pan.getMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGetMapCluster</u> <u>KineucstGe</u>	MapQuest_Directions_netDirections	n an	!	nap				.,
MapQuest-GetMapCluster-Pan.panS MapQuest-GetMapCluster-Pan.panE MapQuest-GetMapCluster-Pan.panW MapQuest-GetMapCluster-Pan.getMapCluster MapQuest-POI.getPOI Pictures SendSMS Weather2 Weather2 Weather3 Stocks FBActions.getFriends FBActions.getFriends FBActions.getFriends FBActions.getFriends FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 SearchAll Search	MapQuest-GetMapCluster.Pan.panN			Activ	ation	Ontic	ons	
MapQuest-GetMapCluster-Pan.panE MapQuest-GetMapCluster-Pan.panW MapQuest-GetMapCluster-Pan.getMapCluster MapQuest-POI.getPOI Pictures SendSMS Weather2 Weather Movies Stocks FBActions.getFriends FBActions.getFriends FBActions.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM FBACTIONS.getIM	MapQuest-GetMapCluster-Pan.panS		r			- p		7
MapQuest-GetMapCluster-Pan.panW MapQuest-GetMapCluster-Pan.getMapCluster MapQuest-POI.getPOI Pictures Send SMS Weather2 Weather Movies Stocks FBActions.getFriends FBActions.getIff FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search	MapQuest-GetMapCluster-Pan.panE		1	••••••			·	1
MapQuest-GetMapCluster-Pan.getMapCluster <u>Kineutaticalitations</u> MapQuest-POI.getPOI Pictures SendSMS Weather Weather Weather Movies Stocks FBActions.getFriends FBActions.getIf FBActions.getIM FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search Ringtones	MapQuest-GetMapCluster-Pan.panW			Cona	rato I	II OB	iacte	
MapQuest-POI.getPOI         MapQuest-POI.getPOI         Pictures         SendSMS         Weather2         Weather         Movies         Stocks         FBActions.getFriends         FBActions.getFriends         FBActions.getIf         FBActions.getIf         FBActions.getIM         FBActions.getIM         FBActions.uploadPhoto         Logon         WebSearch         RSS1         AllTop10.getAllTop10         SearchAll         Search         Starter         Starter <td< th=""><th>MapQuest-GetMapCluster-Pan.getMapCluste</th><th>r 📳</th><th>'</th><th>Gene</th><th>raie e</th><th></th><th>yout</th><th>•</th></td<>	MapQuest-GetMapCluster-Pan.getMapCluste	r 📳	'	Gene	raie e		yout	•
MapQuest-POl.getPOI       Share Web Componen         Pictures       If         SendSMS       If         Weather2       If         Movies       Stocks         FBActions.getFriends       FBActions.sendIM         FBActions.sendIM       FBActions.uploadPhoto         Logon       WebSearch         RSS1       AllTop10.getAllTop10         RingTone-Top10       SearchAll         Search       Search         Search       Search	MapQuest-GetMapSingle.getMapSingle				14			
Pictures Share Web Componen Send SMS Flactons.getFriends Stocks Stocks Stocks Stocks Stocks Status Stocks.sendIM Status	MapQuest-POI.getPOI			<i>/</i>				
Send SMS Weather2 Weather2 Woather Movies Stocks FBActions.getFriends FBActions.setStatus FBActions.setStatus FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search	Pictures			Snar	e wet	Cor	npor	ien
Weather2 Weather Movies Movies Stocks FBActions.getFriends FBActions.setStatus FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search Search	SendSMS				Г			
Weather Movies Stocks FBActions.getFriends FBActions.setStatus FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search Search	Weather2							
Movies Stocks FBActions.getFriends FBActions.sendIM FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search Ringtones	Weather	<b>H</b>						
Stocks FBActions.getFriends FBActions.sendIM FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search-Ringtones	Movies							
FBActions.getFriends FBActions.setStatus FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search Search	Stocks	. 13						
FBActions.setStatus FBActions.setItus FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search-Ringtones	FBActions.getFriends		-					
FBActions.sendIM FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search-Ringtones	FBActions.setStatus							
FBActions.getIM FBActions.uploadPhoto Logon WebSearch RSS1 AIITop10.getAIITop10 RingTone-Top10 SearchAII Search Search-Ringtones	FBActions.sendIM	- <b>1</b>	•			s,		
FBActions.uploadPhoto Logon WebSearch RSS1 AllTop10.getAllTop10 RingTone-Top10 SearchAll Search Search Ringtones	FBActions.getiM				i na internet	e v 🖗 -	2. A	
Logon WebSearch RSS1 AllTop10.gelAllTop10 RingTone-Top10 SearchAll Search Search-Ringtones	FBActions.uploadPhoto							
WebSearch RSS1 AllTop10.gelAllTop10 RingTone-Top10 SearchAll Search Ringtones	Logon							
KSS1 AllTop10.gelAllTop10 RingTone-Top10 SearchAll Search Ringtones	wedsearch		• • •					
All lop tu.getAll lop tu Ring Tone-Top 10 Search All Search - Ringtones	KSS1 ·							
Ring Fone-Foptu SearchAll Search-Ringtones	Alliopiu.gelAlliopiu							
Search Search-Ringtones	Kingtone-roptu Seereball							
Search Search-Ringtones	Searchau							
Search-Kingtones	Search Directories							
	Search-Kingtones	•						

## E: Completing a Web Component with the "Generate UI" Option.

When the Generate UI Objects Dialog Box first appears any required Generated Inputs and Outputs are pre-registered in the "Generated" Inputs and Outputs Lists. All optional candidates, if any, are placed in the "Candidates" Inputs and Outputs Lists. All Inputs and Outputs are defined by their "Symbolic Field Names", which will, during this process, finally be bound to auto-generated Xpressmo UI Objects.



a. Comments:

As an Input or Output Object is selected from any of the four lists below, the Comment, retrieved from the Web Component, is displayed in the Comments field on the top for informing the Author about the meaning of that Object.

b. Select Input Page

This is set initially to the current page of the application. It can then be reset to any other Page. In Phase 1 there is a restriction of only one Input Page.

c. Select Results Page

This is set initially to the "Select Results Page" from the Add Web Component Dialog Box. It can then be reset to any other Page. In Phase 1 there is a restriction of only one Output Page.

d. Input Candidates

Any Optional Input Candidate can be selected and, when the "Generate Input" Button is touched, placed into the Generated Inputs List. Note that as each Candidate is selected the Comment related to it is displayed.

#### e. Generated Inputs

Any Generated Input can be selected and, when the "Remove Input" Button is touched, returned to the Optional Inputs List, unless it was a required Input.. Note that as each Generated Input is selected the Comment related to it is displayed.

f. Output Candidates

Any Optional Output Candidate can be selected and, when the "Generate Output" Button is touched, placed into the Generated Outputs List. Note that as each Candidate is selected the Comment related to it is displayed.

\$

#### g. Generated Outputs

Any Optional Output Candidate can be selected and, when the "Generate Output" Button is touched, placed into the Generated Outputs List. Note that as each Candidate is selected the Comment related to it is displayed.

h. OK

When the OK Button is touched the Dialog Box will close and all the Items listed in the Generated Inputs will appear on the "Input" Page and all the Items listed in the Generated Outputs Lists will appear on the "Results" Page.

Using the example of MapQuest (Get Single Map), the graphic below shows the state after the Optional Input for Street and the optional Outputs for Latitude, Longitude and a Single MapImage have been selected.

		a a second and a second a second a second a	
ctivation Event	Select the input Page		Select the Result Pag
ire 💌	Define Location		Map
put Candidates	Generated inputs	Output Candidates	Generated Outputs
at	zip	getmapSingle-MapURL	Lat
ng	height		Lng
	width		getmapSingle-Mapima
	Submit		
	street		
			( management
Generate Input	Remove input	Generate Output	Remove Output
		International Action of Contractor	

1. The Input Page is shown below. Note that:

There are 5 Text Fields lined up 2 per line.

- 2. There is a Submit Button, that when selected, shows under the "Bindings" Tab the the Web Component Instance "1: Input\_Maps" has been assigned.
  - a. The Web component instance Naming Convention incldes three components:
    - i. The Number of the Instance (This permits multiple unique assignments of the Same Web Component to the same input page)
    - ii. The Input Page Name (This permits multiple unique assignments of the Same Web Component to multiple Input Pages,
    - iii. The Name of the Web Component.
  - b. The "Edit" and "Remove" Buttons are now active.



After the UI is generated the author can make any changes to the the UI objects that is desired. There are no known restrictions. The generated results page is shown to the right. The Latitude and Longitude fields, and how they are utilized, will be discussed later.



If the "Edit" Button is touched under the Bindings Tab then the Dialog Box below will appear. All the bindings of the Data Fields defined by the Web Component are now bound to the generated UI Objects. Any required Inputs or Outputs are identified with a light red background. From here the author could choose to select other UI Objects to replace the ones that had been automatically generated, or to add other optional UI Objects. If any generated required UI Object had been deleted a warning message will appear when the dialog box appears. The designer can then select any other UI Object that had been defined that meet the conditions of the hinting as defined by the Web Component Registry Administrator.

isparit.		enne en la constance de la constante de la constance de la constance de la constance de la constance de la cons	
<b>Ip</b>	height	width	mapScale
Define Location zip	Define Location_heigi	Define Location widt	Define Location map -
Submit	street	Lat	Lng
Define Location Subr -	Define Location_stree	• None •	None -
at a second and a second and a second a At	Lng	getmap Single-MapImage	getmapSingle-MapURL
at 🚽	[Lng	getmap Single-MapIm      Activation E	None
		Fire	<b>T</b>
Web Component Mana	iger URLs Disca	rdable Controls Results P	age
http://localhost8080/wcm	/playerreg - Instance	of Application - Map	<ul> <li>OK Cancel He</li> </ul>

Required Inputs can be assigned to Persistent Variables either manually or automatically (see Section F).



The activation event can be mapped to a Page Entry Event or any of 5 Navigation Actions.

The designer can also change the Server that will be used for the Web component Manager. The default will be the Server that is currently selected for the Publishing Platform.

For Express Mobile designers the current available server choices are:

The Express Mobile QA Server: "http://qa.xpresmo.com/wcm/playerreq"

The Express Mobile Dev. Server: "http://dev.xpresmo.com/wcm/playerreq"

The Staging Server: "http://stg.xpresmo.com/wcm/playerreq"

The Public Server: "http://st.xpresmo.com/wcm/playerreq"

Local Host: http://localhost:8080/wcm/playerreq"

The currently selected Server that the Publishing Tool is logged into will be the default.

Discardable Controls are not implemented in Phase One but are defined in the PDL (See Appendix B).

There are a series of rules that the Publishing Client follows for this automatic generation.

Input Object Generation:

1. Text Fields (Single Line Alpha or Numeric Fields)

Text fields are generated first, with the rules:

- a. If there are more than 2 then there will be 2 per line, otherwise they will span the width of the page, with left and right margins of 10 pixels.
- b. The height will be calculated based on the selected Font from the currently selected Object Style.
- 2. Text Areas (Multiple Line Alpha Fields)

Text areas are generated second, with the rules:

- a. Two Lines will be visible.
- b. The Object will span the width of the page, with left and right margins of 10 pixels.
- c. The height will be calculated based the number of visible lines and on the selected Font from the currently selected Object Style.
- 3. Check Boxes will be generated next with the following rules:
  - a. All Check Boxes will appear on the same line, with a size of 20 by 20 pixels.
  - b. There will be a left margin of 10 pixels.
  - c. Each Check Box will be separated by 20 pixels.
- 4. Lists will be generated next with the rules:
  - a. Three Items will be visible.
  - b. The Object will span the width of the page, with left and right margins of 10 pixels.
  - c. The height will be calculated based the number of items and on the selected Font from the currently selected Object Style.
- 5. Last will be the Submit Button, if the Submit Function has not been assigned to another Input.

The rules will be:

- a. The String "Submit " will be the default string.
- b. The Standard HTML Form Colors will be applied.
- c. Submit Button will be placed in the lower right hand corner of the visible page with a right margin and bottom margin of 10 pixels.
- 6. All Lines will be separated by 5 pixels.
- 7. The Object Names will be the same as the Symbolic Field Name as defined by the Web Component.

**Output Object Generation:** 

- 1. If the "Results" page is the same as the "Input Page", then the Output UI Objects will appear directly below the last Input UI Object Line, otherwise it will start with a top margin of 20 pixels.
- 2. Text Fields (Single Line Alpha or Numeric Fields)

Text fields are generated first, with the rules:

- a. The Object will span the width of the page, with left and right margins of 10 pixels.
- b. The height will be calculated based on the selected Font from the currently selected Object Style.
- 3. Text Areas (Multiple Line Alpha Fields)

Text areas are generated second, with the rules:

- a. Two Lines will be visible.
- b. The Object will span the width of the page, with left and right margins of 10 pixels.
- c. The height will be calculated based the number of visible lines and on the selected Font from the currently selected Object Style.
- 4. Check Boxes will be generated next with the following rules:
  - a. All Check Boxes will appear on the same line, with a size of 20 by 20 pixels.
  - b. There will be a left margin of 10 pixels.
  - c. Each Check Box will be separated by 20 pixels.
- 5. Images and Slide Shows will be generated next, with the rules:
  - a. The Object will span the width of the page, with left and right margins of 10 pixels.
  - b. The Height will either be 60 pixels if there are multiple UI Objects defined for the Result Page, or the Height will span to the bottom of the page.
- 6. Videos will be generated next, with the rules:
  - a. The Object will span the width of the page, with left and right margins of 10 pixels.
  - b. The Height will either be 60 pixels if there are multiple UI Objects defined for the Result Page, or the Height will span to the bottom of the page.
- 7. Lists will be generated next with the rules:
  - a. Three Items will be visible.
  - b. The Object will span the width of the page, with left and right margins of 10 pixels.
  - c. The height will be calculated based the number of items and on the selected Font from the currently selected Object Style.
- 8. All Lines will be separated by 5 pixels.
- 9. The Object Names will be the same as the Symbolic Field Name as defined by the Web Component.

xpressmo.com

#### IPR2021-01146 Page 00395

## F: Completing a Web Component with Existing UI Objects.

The example below is adding a Yahoo Search Web Component with bindings to previously created UI Objects.



Upon touching the "OK" Button the graphic below will appear. Note that "in1", "Submit" and the "Control List: Title" are all populated with an existing previously created UI Object, while all the other fields have "None" selected.

na e na sun i <sup>t</sup> ne e frankeitett formanietiken.					and a second	sinaa sinaa muusiya
in1	in2		Submit			
Search_in1	None		Search_in1		•	
totalresultsAvail	totairesults	Returned	Alt_Title		Control List: Tit	e
None	None	<u> </u>	Alt_Title	•	Title	•
					Assemble	List
URL	ClickURL		DisplayURL		Summary	
urf	None	٦	None	Activation Ev	Summary rent	•
		<b>D</b> <sup>1</sup>		Fire	<u> </u>	
web Component Ma		Uiscardad	He CONTOIS	Results Pa		Cancol Hat
As with the MapQuest Web Component example, there are both required and optional input and or output symbolic fields. In this case there is a required Symbolic Objects as follows:

- An Input object for entering in the Search String.
- An Output Control List which will display the Search Result Titles (See Section I).

Color Coding in the graphic above has the following meanings:

Required Input or Output

Control List

Child Output Object

All other symbolic fields are optional.

The rules for selecting which existing UI Objects should be made available to a given Symbolic Field is based on the web Component Definition and the hinting that was applied by the Web Component Registry Administrator.

Section B shows all of the currently supported UI Objects for Xpressmo's Web Component. These will be mapped to a Symbolic Field Name based on the Supported Data Types shown below.

- 1. "Boolean"
- 2. "Int"
- 3. "String"
- 4. "multilineString"
- 5. "ImageURL"
- 6. "VideoURL"
- 7. "List" (Contains a parameter that is converted to an Index)
- 8. "ComplexList"
- 9. "Slideshow"
- 10. "SearchResponseList"
- 11. "RSSList"
- 12. "SingleSelectionList"
- 13. "MultiSelectionList"

- 14. "ServiceActivation"
- 15. "ChannellmageURL"
- 16. "ChannelDescription"
- 17. "ChannelTitle"
- 18. "URL"
- 19. "Audio URL"
- 20. "Purchase URL"
- 21. "ImageData"
- 22. "ImageListData"
- 23. "PersistentVariable";
- 24. "PipelineMultipleSelect";
- 25. "PhoneNumber";
- 26. "HiddenAttribute";
- 27. "CollectionList";

r

Data Types	Preferred Input	Input Candidates	Preferred Output	Output Candidates
boolean	Check Box	Check Box	Check Box	Check Box
Int .	Text Field (integer)	Text Field (integer) Text Field (Phone #) Text Field (SMS #) Choice List (single select)	Text Field (integer)	Text Field (integer) Text Field (Phone #) Text Field (SMS #) Choice List (single select) Text Button
String	Text Field (Alpha)	Any	Text Field (Alpha)	Any
multilineString	Text Area	Text Area	Text Area	Text Area Paragraph
ImageURL	N/A	N/A	Image	Image Slide Show
VideoURL	N/A	N/A	Video	Video Slide Show
List	Single Item List	Single Item List Multi-Select List Complex List Choice Slide Show	Single Item List	Any List Type Any Choice Type (see Complex List Specification)
ComplexList	Complex List	Single Item List Multi-Select List Complex List	Single Item List	Any List Type (see Complex List Specification)
Slideshow	Slide Show	Slide Show	Slide Show	Slide Show
SearchResponseList	N/A	N/A	Search Response List	Search Response List Control List Complex List Choice
RSSList	N/A	N/A	RSS Display List	RSS Display List Control List Complex List Choice
SingleSelectionList	Choice	Choice Complex List	Choice	Choice Complex List
MultiSelectionList	Multi-Selection List	Multi-Selection List	Multi-Selection List	Multi-Selection List
ServiceActivation	Submit Button	Any	N/A ·	N/A
ChannelimageURL	N/A	N/A	lmage	Image Video Slide Show
ChannelDescription	N/A	N/A	Text Area	Text Area Paragraph Text Field Text Button List Choice

The Table below shows the mapping of Xpressmo UI Objects to the Supported Data Types.

.

ChannelTitle	N/A	N/A	Text Field	Text Field
				Text Button
				Paragraph
				Text Area
			ı	List
				Choice
URL			Text Field	Text Field
			(URL request)	(URL request)
Audio URL .			Text Field	Text Field
			(Audio URL request)	(Audio URL request)
Purchase URL			Text Field	Text Field
			(Purchase URL request)	(Purchase URL request)
Image Data			Image	Image
				Slide Show
Image List Data			Slide Show	Slide Show
				Image
Persistent Variable	N/A	N/A	N/A	N/A
Pipeline Multiple Select	Multi-select List	Multi-select List	N/A	N/A
		Complex List		
		Slide Show		
Phone Number	Text Field	Text Field	Text Field	Text Field
	(numeric type)	Text Button	(numeric type)	Text Button
Hidden Attribute	Complex List	Complex List	Complex List	Complex List
		Slide Show		Slide Show
Collection List	N/A	N/A	Slide Show	Complex List
				Slide Show

### G: Persistent Variables.

When editing a WebComponent in the publisher, any persistent variable that has been defined will appear as a choice for any input. The variable name will be preceded by a lozenge character to differentiate it from a UI Object Name.

If selected, then the UI Object that had previously been assigned to that Symbolic Field will be released and can now be deleted..

In the player when this page is entered, and if the persistent variable has been defined, the WebComponent will be executed using the persistent variable value as an input.

Persistent variables can be used in many ways to make the the UI of the Web Component more efficient and intuitive.

1. Utilizing login or PIM data:

Often the application was, when first activated, a login page so that the user could register themselves for various web component services or communication functions. This data, if placed in a persistent variable, could then be an input to any web component that, as an input, uses location, phone number, name, password, etc. automatically.

2. Improving the UI:

Very often the output from a web component is a URL that will execute a particular web service, such as previewing a product to be purchased, or initiating the purchase transaction itself.

An Image, that clearly labels the web service that could be initiated, could share the same persistent variable as the Text UI object that was assigned to the Web Component output that received the URL. The Text UI object could be hidden so that the UI is more attractive and intuitive.

3. Integrating Web Components.

A persistent variable can be utilized in three different ways.

- a. As a direct input to a WebComponent.
- b. Assigned to a UI Object that is selected as a Web Component Input
- c. Assigned a UI Object that is selected as a Web Component Output

Each of these uses of Web Components can come in handy for many use cases.

A few examples are:

1. Have Web Component Output Values become inputs to a different Web Component.

In this case assign a persistent variable to the output UI Object. Upon completion of this Web Component the persistent variable will be updated to the new value returned. This persistent variable can now be referred to as an input to any other Web Component, no matter what pag it originates from.

2. Set Defaults at Authoring time for a Web Component.

The author could set a default input value to a hidden UI object, and then assign a persistent variable to it. When the Web Component is first invoked it will use this value.

3. Mashing Up Outputs from Multiple Web Components

Since any output from any web component can be assigned a persistent variable, the author can now link these persistent variables, as desired, to any UI Object, any relevant Web Component Input, or any combination of the two.

# H: Enhancing Web Components with Auto-generated Persistent Variables.

Let's work with the Web Component MapQuest-getMapCluster. It is a more powerful Web Component then the MapQuest-getSingleMap used in Section D.

1: First, auto-generate the UI objects.

Activation Event	Select the Input Page	Preload Responses	Select the Result Pa
nput Candidates	Generated inputs	Output Candidates	Generated Outputs
Lat Lng	Zip height width mapStale Submit street	**************************************	mapuri menSentafortikita mapimage Lat Lng

Fig 11-14

Fig 11-15

2: Then Edit the Web Component.

(P) Edit WebService Binding	11Define1location1MapQuest-	GetMapCluster-Pan.getMapClu	ister
zip	height	width	mapScale
Define Location Szip 🖉 🗸	Define Cocation <u>1</u> heigi 💌	Define Location Widtl -	Define Location≝mapi ▼
Submit	street	Lat	Lng
Défine Locétion_Subr -	Define Location_stret -	Nona	None –
mapuri	map ScaleforURL	mapimage	Lat
mapuri	map ScaleforURL	mapimage	JLat
Lng . Lng 🛃	an an tha an	Activation Ev	rent Preload Responses
Web Component Manay	ger URLs Discardab	le Controls Results Pa	ge
http://localhost:8080/wcm/	playerreg V Instance of A	pplication <b>-</b> Map	OK Cancel Help

Note: The Submit Symbolic Field is still assigned to a UI Object.

3: Now set the Submit Symbolic Field to "None"

Maria Managaria	zip Define Location zip	height Define Location heig -	width Define Location Lwidtl -	mapScale Define:Cocation≚map →
	Submit	street Define Location_stree	Lat None	Lng None
1111/1/ PARLING TRUTTING AND AN ADDRESS OF COMING ST	mapuri mapuri	map ScaleforURL	mapimage mapimage	Lat

#### 4: Touch "OK".

A message appears informing the User that all the Input UI Object, have now been assigned a persistent variable.

The Web Component itself has now been attached to the Results ("Map") page and will be executed when that page is entered. Thit is to say, the Submit Button on the "Define Location Page has had the Web component removed from it, but if you go to the "Map" page, and with the Page selected (no page object selected) and touch the "Bindings" tab, the Web component will be available for further editing.



Touching the "Edit" button will confirm the operation as shown below.

ip	height	width	mapScale
izip	āhēight -	¤width ↓	emapScale -
Submit	street	Lai	Lng
vone 👔 🕹 🔽	□street _	None 🗾	None
napurl napurl	mapScaleforURL	mapimage	Lat
00			
.ng		Activation E	vent Preload Response
Web Component Manag	er URLs Discarda	Fire File Results P	age Not Used
http://localhost:8080/wcm/c	laverreg - Instance of	Application	- OK Cancel He

Note that all the inputs now have the lozenge character preceding them to indicate that they are bound to a persistent variable.

Selecting one of these objects such as the "zip" text field will also show this in the Resource Inspector Panel.

Object	Name	
zip		
Left 120	Top 10	Width 100
Attribu	ites	Draw
Defau	it	- ŀ
Rememi V	ber Me	
Add Per	sistent V	ariable
zip		
		Fig 11-18

Discussion:

We can divide web components into two very distinct categories:

1. Web components that are activated on a Page Entry Event:

These type of Pages can be entered from any other page and still yield the same result. Pages using these types of Web components make ideal initial pages for Web component Widgets.

2. Web Components that are activated by a Navigation Event (FIRE, TOP, RIGHT, DOWN, LEFT):

These types of Web Components are best utilized when the activation page and the results page are the same. They make the Web Component interactive.

Although it is possible to create near identical results in both cases, there is great utility in having a Web component, whose inputs are all bound to UI Objects that have attached persistent variables, be the initial Web Component that is executed when the page is first entered. This means that all of the Web Components that might be defined to work within that page will always have symbolic access to the initial set of parameters that were associated with the initial Page Entry Web Component.

### I: Restrictions for Editing UI Objects related to Web Components:

There are three general cases in which the editing of a UI object will be restricted.

1. Delete a UI Object that has a Web Component Attached.

In this case the following message will appear and deletion will be suppressed.

Dig Object getPOI cannot be deleted!	(A) 8,00000
	2.000
Object getPOI has 1 attached Web Components.	
 Touch Bindings Tab and Delete the Web components First.	100
 OK	WEAR AND A
Fig. 1.1	0

ż

If the author's intent is to reassign the Submit function to another UI Object then first create that UI Object, then Edit the Web Component and reassign the Submit Symbolic Function to the new UI Object. The original UI Object can now be deleted.

2. Delete a UI Object that is bound to a Web Component Input or Output or delete a UI Object that is attached to a persistent variable that is bound to a Web Component Input.

STATES.	12 Object zip cannot be deleted!
ľ	
	Object zip is attached to the Web Component 1: map_MapQuest-GetMapCluster-Pan.getMapCluster
	Touch Bindings Tab, Edit, and reassign zip first.
	<u>ok</u>

Fig 11-20

If the author's intent is to reassign the Web Component Symbolic Field to another UI Object then first create that UI Object. Then touch the "Bindings" tab related to the UI Object (or Page) that has the identified Web component attached. Touch "Edit" and then either reassign the Symbolic Field Function to the new UI Object or set that function to "None". The original UI Object can now be deleted.

3. Remove a Persistent Variable from a UI Object in which that Persistent Variable is bound to a Web Component.

🔺 <u>Rem</u> e	mber Me	Refere	nce for n	napzip	cannot be	deleted			• •	
:	- 14			- 12	the start		i (a se	1	1.1	
Object n	napzip's I	Remen	iber Me I	Refere	nce zipco	de is attac	hed to	a We	b Com	ponent.
This Ref	erence m	nust be	change	d in the	e related \	Neb Comp	onent F	irst.		
						3. S.			a se se	
ACT: 017422045-020	and the second secon		M ACT ALL &	1 ayu ya katalata			al a constante	ing the second second	an aire an an aire an a	UK
										Fi- 1

If the author's intent is to reassign the Web Component Symbolic Field to another UI Object then first create that UI Object. Then touch the "Bindings" tab related to the UI Object (or Page) that has the identified Web component attached. Touch "Edit" and then either reassign the Symbolic Field Function to the new UI Object, reassign it to a different Persistent Variable or set that function to "None". The original UI Object's Persistent Variable binding can now be removed..

### J: Working with Control Lists:

The following Object Types are variations of Control Lists:

- Complex Lists
- RSS Display List
- Search Response List

The Control List Model definition includes any number of subordinate (Child Output Objects) whose contents are known at all times to the Control List, and the Control List manages the appropriate display of their contents in two very different ways.

1. Display as independent UI Objects:

Each Child Output Object is visible, and may or may not be selectable. As the user navigates down through the Control Lists, the values for each of the child lists, related to that item, is displayed. An example is a RSS Display List where the Summary is displayed as a child Object as the Title for the RSS Feed, is selected within the Control Lists.

2. Display as a Mash up within the Control List:

Instead of displaying the attributes of each child object in a separate UI object, that attribute is displayed as part of the Item of the Control List. Usually the Child Output Object is set to "hidden".

It is possible to employ both techniques within a Control List implementation.

Fig 11-22

#### Publisher Interface

#### A: UI Implementation

If a Web component has a Control List, it will be a required Output field. The example to the right is the ThumbPlay AllTopTen Store Front Web Component. The automated feature "Generate UI Objects" was selected.

Note: Control List will be identified by the prefix "Control List".

Select the	nput Page	Preload F	Responses	Select the	Result Page
default	-	Not Used	<b>•</b>	default	-
Input Candidates	Generated in	puts	Output Candidat	89	Generated Outputs
calegoryNames pname Submit			All_name albumID smatlipg contentiD albumTitle artistName ringtonePreview mobilePurchase	v eURL	calegory Control List name
Generate Input	Remove	Input	Generate Outp	aut	Remove Output
					OK Cance

Note: In the example below the Child Output Objects:

- smalljpg
- albumTitle
- artistName
- ringtonePreview
- mobilePurchaseURL

have been selected for generation.

-						
	Select the t	nput Page	Pretoad I	Responses	Select the	Result Page
	default	-	j Not Used		jaetault	<u> </u>
input Candida	les	Generated	Inputs	Output Cand	idates	Generated Outputs
categoryNam	es	<b></b>		Alt_name		category
pname				albumiD		Control List: name
Submit				contentiD		smalljpg
		1				albumTitle
				1		artistName
						ringtonePreview
						mobilePurchaseURL
		1				
			:			
ar	. 10 A	I		I		l
	. 1	D		Constato	Sutnut 1	Domesic Output



The page view to the right shows the results of the generated outputs after a bit of editing. The Output UI Objects that were generated or modified with persistent variables are:

Output Object	Default UI Object	Artist	1
Control List: name:	Complex List	Album:	; ; ;
Smalljpg	Image		, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
albumTitle	Text Field		¥ ¥ ¥
artistName	Text Field	Gel	Ň
ringtonePreview	Text Field		
mobilePurchaseURL	Text Field		\$ } ;
<u></u>		Fig	11.2

The first four objects continue to use the default generated UI objects. The mobilePurchaseURL was initially generated as a Text Field, but an Image Icon "GetIt" was placed on the page and then a subsequent editing of the Web Component switched mobilePurchaseURL to the Image.

Note: This could also have been accomplished by assigning the same "Remember Me" persistent variable to both the generated Text Field and the lcon Image. Hiding the Text Field would then effectively switch the Purchase function to the Image lcon.

Control Lists have a special property that is any of the following object types are assigned to any of its Child Output Objects, and the Events for the Control List are not overridden, then the function assigned to these child objects will be executed when the Item selected in the List is fired on.

These Object Types (in precedence order) are:

- HiddenAttribute (Web component Specific Function)
- PhoneNumber (To place a phone call)
- AudioURL (to stream, if supported, and play an Audio File)
- URL (to WAP or Web Browser Page)

Since the ringtonePreview is of the object type AudioURL, then the current URL assigned to that control List Item's ringtonePreview Field will automatically be played on a fire event.

If the author wishes for a particular FIRE event to override any of the auto-generated events as described above, and for that choice to take precedence, then follow the following steps:.

1. First touch the Events Tab when the Control List bounded Complex List is selected.



2. Then Select an Events or Web Services action. When selecting the following warning will appear.

		Events and Web Services None
<ul> <li>Override Werning</li> <li>Warning! About to Overrid</li> <li>Click OK to apply your selected E</li> </ul>	e Item Exit Events. xit Event to all items.	Goto External Web Page replacing Current Frame Goto External Web Page Launched in a New Window Goto a specific Page View State State Goto External Web Page replacing the Top Frame Goto the next Page View Evolta Lange of the lock
् ् enter artist, song or keyword		PauseResume Page Timeout PauseResume Page Timeout Execute an Application Goto a Specific Sikle in a Page View Exit Application
Jollipop 📲		Exit Player Place Phone Call Send Stribu an EPE

3. Touch OK and complete the Selection Process. This setting will now be executed when firing on any of the Control List's Items.

Green:

Red:

Blue:

Black:

When Editing a Web Component, as shown to the right, the class of output type is shown with a color coding methodology.

Edit Web Service Binding 1: default Allfop10.getAllTop10 豪 \* 1.2.2 categoryNames pname Submit None -None -• None albumiD Alt\_name Control List: nume J • name None ... category None 4 Collection List Assemble List albumTitle artistName smalling contentiD Control List Smallimage -None albumTitle artistName • -**Child Output Object ringtone**Preview mobilePurchaseURL ingtonePreview mobilePurchaseURL -Preload Responses Other Not Used Web Component Manager URLs Discardable Contr **Results** Pac OK Cancel Help Instance of Session http://jocalhost8080/wcm/playerreg default Fig 11-27

If there is a Control List assigned to the Web Component, then an "Assemble List" button becomes available.

Firing will open the Assemble Complex List Dialog Box. All available Child Objects under the control of the Control List can now be selected and placed in any of the following places within the Control List:

- First Line Body
- Second Line Title .
- Second Line Body

If a second line text attribute is selected then the item layout for all items in the control list will each have two lines.

Select Output	Select Control List String
urtis dName 🔭 🐭 🗸 🗸	lst Line Body -
1st Line Title	artistName
2nd Line Title	2nd Line Body
	Remove OK Cancel Het

1st and 2nd line Titles will use the Control List's Text Font and Color.

The 1st and 2nd line Body Text will use the Item Font and Color.

Touching the "Remove" button will cause the Child Object's value to be removed from the Item.

Publisher Interface

The choices made in the Assemble Complex List Dialog Box will be reflected in the List Attributes tab of the Resource Inspector.

The settings for activating the Titles and Strings are only available through the Assemble Complex List Dialog Box. The settings for activating Icons are not currently supported.

There are several additional controls that are available for Control Lists that will affect its appearance.

Either Line can be right justified.

The List Item Typeface can be changed.

The List Color can be changed.

Object Attributes	. List Attributes	Item Attri	butes
List ttems	List Styles	List Iten	n Layout
<b>.</b>	default -	I 1st Line:	2nd Lin
Add Item	Leading	) Icon: 🗖	Г
Remove Item	Line	e Trile: 🔽	Г
Remove All	Use All	l. Fon <b>t: 🦵</b>	Г
	Bod	y Icon: 🗖	Г
List Selector Type Web Component Contro	of List - Body	String: 🔽	Г
	Use Ali	t.Font: 🦳	
	Right J	ustify: 🦳	٣
i en ser	Alternate Font Nan Arial	ne: Alternate i Bold	ont Styl
	Second Line Inden	t Define Sec	ond Col

For course, the Object's Text Font and color can be changed under the Object Attribute tab. Any change to the Font Size will be automatically be adopted by the List Item Font.

Chapter 11.37

#### **B:** Behaviors

The graphic to the right shows how the Control List will be drawn based on the setting as shown in the example below. When the Item is selected the Object and List Item fonts will display but the Text Color will be white and the background blue.



Fig 11-30

### K: Working with Collection Lists:



Either Slide Shows or Complex Lists can be assigned to a Collection List although the default will be a Slide Show. There will be a symbolic reference to each element of the Slide Show or Complex List to a Control List definition, which will also include and the attributes for each Control List set of Child UI Objects.

Fig 11-33



In the example below, the Tabs are actually a Slide Show, as shown in the "Working With Slides" Dialog Box to the right.



Touching the Right Arrow Key will cause the Slide Show to advance one slide. As each slide contains a symbolic reference to a different Control List definition the Control List is repopulated with that definition.



Fig 11-34

At any time, touching the down arrow key will cause a navigation event to first select, and then navigate to the next Item in the Control List as shown to the right.

If a Right or Left Arrow Key is pressed while navigating through the Control List the Collection List moves to the next element (Slide). This Slide's symbolic reference tp a Control List definition is then activated and the Control List is repopulated as expected.

### Tadi 🖉 🔤 AS UBEMORE ringtones video graphics games lames Blunt is Beautiful Musical Karme Say Hello To My Little Friend Scarface (I et Like Me David Banner ft. Chris Bro U And Ur Hand Pink White & Nerdy Weird Al Paul Pierce Flies For Slam Boston Celti Rider (Pt. 2) G-Unit ् enter artist, song or keyword Exit Fig 11-35 भिनेत्री 😳 🖾 AS UBEMORE ringtones video graphics games Spain Soccer Ball America The Great Fireworks Patriotic Fireskull Wrapped Up (Ashanti) Ashanti Black impala Lowrider (310 Motoring) USA Soccer Ball Psychedelic Box ्, enter artist, song or keyword

e¢a) Fig 11-36

Chapter 11.39

### L: Collection List Keys and Complex List Indexes

#### 1: Collection Lists:

Collection Lists can be auto-generated or manually created. If automatically generated, then their keys are generated and placed as hidden attributes for each Slide (if a Slide show) or each List Item (if a complex list.

For example, using the Thrmbplay AllTopTen Store Front Web Component, if the Web Component was autogenerated, then a Slide Show would be generated with a Slide for each key. This information was retrieved from the Model Definition as shown below.

ServiceMapping="//ns2:contentItemResponse/contentItemRequest/moduleCode" name="category" dataType="CollectionList" use="required"> <Comment>The category like ringtones, video, games, graphics</Comment> <ListValues value="ringtone-chart">RingTone Chart</ListValues> <ListValues value="video-chart">Video Chart</ListValues> <ListValues value="video-chart">Graphic Chart</ListValues> <ListValues value="graphic-chart">Graphic Chart</ListValues>

The generated Slide Show will start with a blank slide for each key. The author can then assign the imagery they wish to visualize the meaning of each Control List.

			Import	Slide Setting:
ringtones įvideo. įgraphics, (games)	(2 Schented Slide Delay(Sec.) in Stay 4 - Non Fire Enabled Shri	Sellings	se Photo	Audio
	RSS Feed. Unassigned Media File Name. ringtones_list_back	Define RSS Sour	elete Settirgs	Colors
	0.1     2       Parameter       ringtone-chart       Bubble Help			Help
		OK Cancel H	₽₽ 2 of 4	

Note that the generated parameter is "ringtone-chart" and the author assigned an Image to the slide that has the ringtones tab selected.

This process could have also been done manually by entering each key from the Model to the parameter field for each slide.

1: Complex Lists:

Control Lists may also have hidden attributes which to define one or more values for requesting to the Web Service (as an Input List Data Type) or, as an index for visualization as an Output Data Type of various field values as part of a set of data records.



In the case above, the MapQuest Web Component generated a Complex List for each zoom level that would be supported. The Internal String Data for each item on the list was retrieved from the Model. And defined below and placed appropriately for each list item.

<ListValues value="6000">Street 1</ListValues> <ListValues value="12000">Street 2</ListValues> <ListValues value="24000">City 1</ListValues> <ListValues value="48000">City 2</ListValues> <ListValues value="96000">City 3</ListValues> <ListValues value="192000">County 1</ListValues> <ListValues value="400000">County 2</ListValues> <ListValues value="400000">County 2</ListValues> <ListValues value="400000">County 3</ListValues> <ListValues value="600000">County 3</ListValues>

This process could have also been done manually by entering each index from the Model to the Internal String Data for each List Item.

### M: Sharing a Web Component.

Section H, "Enhancing Web Components with Persistent Variables" described a powerful way to combine, extend and mash up various Web Components to create great Services. However, sometimes the requirement is just to use a single Web Component, but extend it through sharing to make it much more interactive and usable. For example:

Using the example of Yahoo Maps, the graphic below shows the state after the Optional Inputs for Street, City, State, Zip, Image\_Height, Image\_Width and Zoom have been selected.

Participate UI Objects for	Map: 👘 🛪 2 🐝 🖓 👘		
integer: 1 to 12 (default: 6	The zoom level for the map, If a radius is specifie	from 1 (street level) to 12 ( d, this is ignored.	country level).
Select the Location	e Input Page	Select the Re	esult Page
Input Candidates location latitude longitude image_type	Generated inputs state ^ zip Image_height image_width	Output Candidates	Generated Outputs Result
radius Generate Input	Remove Input	Generate Output	Remove Output OK   Cancel  Help

Fig 11-39

The generated "Location" page is shown below.



The generated "Map" page is shown to the right.





- 1. Go to the Results Page ("Map") of this application.
- Add appropriate UI Objects for further interactions with the Web Component. (We might add a Slide Show UI
  Object that would be assigned to the Zoom Symbolic Field Name of a mapping Web Component. In this case
  we will use Yahoo Maps as shown below)
- 3. Touch the "Bindings" Tab and Select "Maps"



Note that "Generate UI Objects" is not available, and "Share Web Component" is available and Selected. When you touch "OK the Dialog Box below will appear.



#### Note:

As seen above, it is possible to assign a Slide Show as a "List" Output Object, so that you can produce graphical controls for various types of Web Components. In addition, there is now an exit event for Slides that permit you to assign "Previous Slide" as well as "Next Slide". This means that functions such as "Zoom In" or "Zoom Out" can be implemented in various interesting ways. Now assign the Submit Symbolic Field and the Zoom Symbolic Field to the Slide Show UI Object and the Location Symbolic Field to the Location Text Field as shown below. Also, since location will be used for the target map, Street, City, State and Zip can now be assigned to "None".

on 👻
on 🔽
on 👻
on 👻
-

After touching "OK" the "Map" the Extended Web component 3:map\_Maps will be assigned to the Slide Show UI Object.



Now save the application and view the results on a phone or in a MIDP 2.0 Emulator.

#### Publisher Interface

#### Chapter 11.45

∑	
  94949	ETCO! Commercial
240  198 Street Level 1	
Street Level 2 Street Level 3 City Level 1	
	Som
	100 t GYANOOIZ008 Data CNAVTEQ2008
Exit Fig 11-45	Exit Merul Fig 1146

Initial View

Result

- Initial mode will be to Zoom in if not on the Maximum zoom level.
- FIRE or, if Enable Fire is selected, the forward and next navigation keys, will Zoom In the Map One Level and move the Slide Indicator.
- If a navigation Key is touched that reverses direction, that will switch the mode to that direction.
- FIRE or, if Enable Fire is selected, the forward and next navigation keys, will then Zoom Out or Zoom In One Level the Map but the Slide indicator will not move.
- The next FIRE or, if Enable Fire is selected, the forward and next navigation keys, will both Zoom Out/In and move the slide Indicator.
- Hitting the maximum or minimum zoom will cause an automatic direction change.



#### Chapter 11.46





## N: How to create a Web Component Model:

See Appendix C.

EXPRESS MOBILE User's Guide

Chapter 12

Working with Object Styles

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 12: Working with Object Styles

A:	Introduction	12.04
B:	Object Styles	12.05

Chapter 12: Glossary

Styles

The underlying set of default appearance attributes or behaviors that define any object that is attached to a style. A very convenient way for quickly creating complex objects, and for changing a whole collection of objects by just modifying their common style.

### A: Introduction:

Object styles are used to define a certain default set of attributes for any given object when it is created. They are extremely useful if you are building pages that have multiple objects and you want them all to have the same attributes. For example, if you are building a Web page and want identical navigation buttons, using this feature will let you click on the style you've defined – font size, color, type, effects, background color, etc. – and all future buttons you add will be identical. You can also modify the style and all existing objects with that style will be changed, even if they are on multiple Web pages. You can also assign a different style to an existing object. This will cause all the attributes of the new style to be assigned to that object.

Page styles are useful if you want the same attributes on each page, including objects. For example, a company Website may want their logo and a navigation bar in the same position as well as the same background color on each page. Using page styles means you can create a style once and each subsequent page added will contain the same objects versus having to create the same style for each individual page while working on this site.

Using these features can save an enormous amount of time – and frustration!

Chapter 12.5

### B: Object Styles:

1: How to Create an Object Style:

Object styles can be defined for any object (text, picture, shape, etc.) and then applied to an object already on the page or to objects yet to be added. There are currently 10 object styles pre-defined.



ited form the currently selected object (as shown to the right with the "gnews01" UI Object selected, or from a previously defined Style Object.



2: How to Modify an Object Style:

To modify an existing object style select the style from the Layout Inspector and touch the Edit Icon. The current definition of the selected style will display. In the example below, the "Title" style has been selected.

,



Change several of the attributes - font name, effects, size, frame style and width and click "ok".

Note that the feedback box shows what the changes look like.

3: How to Save an Object Style:

Touch the Save Icon from the Command Floating Palette or select "Save As..." from the File Menu.

The dialog box to the right will appear.
Once this object style is saved it can be
imported in another editing session as de-
scribed below.

4: How to Import an Object Style from the Styles Library:

To access this style in the future, select

File Name	Consiste OVCA Form Factor
NewStyle	Generate GVGA FORM FACTOR I
File Type	Generate QCIF Form Factor:
Application	Embed Application:
Current Web Page Widget Object	Embed Images/Slides:
Application Template	Embed Audio/Video:
Style Template Selected Object Style 2	
	OK Cancel Hel

"Styles" and then click on "Import", or Select the Import Icon in the Layer Inspector for Object Styles.



The

"Import an Object Style" dialog box appears; select "AlienAttach.spy" and click "OK". This object style will now appear in the "Style" List in the Layout Inspector.

#### 5: Inheritance:

Once you create an object using a style and then change some attribute of the object while onscreen without modifying the style itself – if you go back & then modify the style and change that same attribute to reflect something else, the attribute for that object won't change. For example, you created a button created using a style that had a red font. You then – onscreen (by using "Edit" or the color drop down menu) - change the font color to blue. You then modify the style to a font color of yellow. The font color of the button onscreen will remain blue.
Chapter 12.8

.

xpressmo.com

### IPR2021-01146 Page 00433

EXPRESS MOBILE User's Guide

> Chapter 13 Animations & Timelines

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 13: Animations & Timelines

A:	Introduction	13.04
B:	How to Define	13.06
C:	Child Object Animations & Behaviors	13.09
D:	Defining Object Entry Timelines	13.11
E:	Defining Object Exit Timelines	13.12
F:	Tips	13.13

# Chapter 13: Glossary

**Slide Show Animations** 

Waypoints

The transition animations that can occur when changing from one slide to another.

The verticies for any object that has an animation trajectory path defined. Each waypoint can have numerous attributes and behaviors assigned to it, so that the user experience can be enhanced in significant ways.

Chapter 13.4

### A: Introduction

Text objects, images, videos, slide shows, shapes and form objects (with the exception of complex lists) can have both directions and animations assigned to them. In addition, directions and animations can be assigned to all of these object types that are defined as child objects. Note that Form Objects cannot be assigned as a child object. To animate the object, be sure that the object is selected and select one of the methods below:

- Touch the "Animation" tab of the resource inspector.
- Touch the Edit Icon from the Layer Inspector and select the "Animation" tab.
- Select "Animations..." from Events off the Menu Bar.

As a shortcut, animations can be defined without activating the timeline.

• Delay:

This is the time, in seconds, that defines when the animation should start relative to when the page comes into view.

Direction:

This defines the location of the object over time. The choices vary depending upon the object type. Generally these types of animations will work quite well, even on fairly low end feature phones.

Movement:

This defines the animation effect, and these choices also vary by object type. It is permissible to have both a direction and/or movement assigned simultaneously to the same object.

Duration:



This represents the time that the animation will remain active from start to finish.

• Frames:

These are the maximum number of frames per second that the animation will be drawn. In the event that the device does not have the processing power to perform at this frame rate then the frame rate will automatically be reduced so that the animation finishes within its allotted time.

### Publisher Interface

Animation Audio:

An audio can be assigned to the animation. It is possible to set the audio to play once or to continue to loop until the animation has been completed.

• Animation Cycles:

Any number of animation cycles can be set. This would cause the animation to repeat itself based on the number of cycles chosen.

Custom Zoom %:

Normally a zoom will zoom the object from zero up to 100 percent of its size. It is possible to increase the maximum size up to 10 times its normal size. This feature is not recommended for feature phones.

Chapter 13.6

### B: How to Define

#### 1: Define Form Objects, Text Button or Paragraph Animations

Any of 31 different directions and 18 different animations can be assigned (See Appendix B).

Select either the "Edit" or "Animate" icons as described above. Choose the desired delay time, direction and/ or animation, the time you want the animation to last, the number of animation frames that will be drawn per second, and the number of times the animation should occur. The time for the animation can be set from 0.5 seconds to 1 minutes, the number of frames per second from 1 to 50, and the number of animation cycles from 1 to 50 or to a continuous ("loop") setting.

To choose a custom zoom level for zooming animations, the range can be from zero to 1000 percent of the defined size for the button or paragraph. 1% means it will zoom to a very small level; 100% means it will remain the same size and anything above 100% will zoom larger.

Avoid Cursor: if the object, during its animation, comes in contact with the cursor it is repelled in such a way that it never intrudes upon the cursor's location. When the cursor is removed from the object's trajectory, the animation resumes as expected. During the time that the object is being repelled by the cursor, no animation time lapses, so that when the animation resumes, it continues as if the collision with the cursor had not occurred, with the exception of the delay.

To add an audio track, select one from the Express Mobile Library or add your own (must be in ".au" or ".wav" formats). You can have the audio track play once or loop. An audio stream can also be added.

Click on the "OK" icon to complete the animation. With the exception of the "Custom" animation, the animation will then be executed for confirmation.

### 2: Define Image or Slide Show Animations.

Any of 31 different directions and 26 different animations can be assigned (See Appendix B). The process for defining the animation is the same as for buttons and paragraphs above.

#### 3. Define Shape Animations.

Any of 31 different directions and 28 different animations can be assigned (See Appendix B). The process for defining the animation is the same as for buttons, paragraphs and images above.

- b: Define General Directions
  - 1: Custom Animation

When the "Custom" animation is selected, a "Define Animation Path" button will appear. After clicking on this button, a message saying "Recording Custom track" will display. Drag the mouse across the Canvas to draw the desired animation. A blue line will be drawn to show the animation (see example below). To stop drawing, right click the mouse and release.



Fig 13-3

Immediately the trajectory (all the points on the screen that the object will move through) will disappear. The number of frames per second setting is ignored.

To use waypoints, see the explanation in the "Events" chapter - Waypoints.

#### 2: Multi-Point

The same as Custom, but with straight lines only. To create, click where you want the line to end. To stop drawing, right click your mouse and release.

To use waypoints, see the explanation in the "Events" chapter - Waypoints.



#### 3: Seek Cursor Movement

The object immediately moves towards the cursor. The speed of movement is calculated to be the time that the object would cross the entire Web page within the time set for the animation. When the object reaches the cursor it immediately jumps to its home location and the animation is completed. It is permissible to assign an animation that will execute simultaneously while the object is seeking the cursor.

4: Attach

The object immediately jumps to the cursor and stays attached to the cursor during the time that the animation is alive. When the time for the animation has lapsed the object will jump to its home position. It is permissible to assign an animation that will execute simultaneously while the object is attached to the cursor.

5: Deposit

The object immediately jumps to the cursor and stays attached to the cursor during the time that the animation is alive or until the user presses the mouse button, whichever occurs sooner. If the time for the animation elapses without a mouse click the object will jump to its home position. If a mouse click occurs while the animation is alive then the object is deposited at the location of the mouse click and the animation is terminated. It is permissible to assign an animation that will execute simultaneously while the object is attached to the cursor.

6: Send Home

The object immediately jumps to the cursor and stays attached to the cursor during the time that the animation is alive or until the user presses the mouse button, whichever occurs sooner. If the time for the animation elapses without a mouse click the object will jump to its home position. If a mouse click occurs while the animation is alive then the object immediately jumps to its home position and the animation is terminated. It is permissible to assign an animation that will execute simultaneously while the object is attached to the cursor.

7: Hover

This is used when you want to play an audio track for a specified period of time for a specific object. The object will hover (remain stationary) and not perform any animations.

### c. Define General Animations

Most are self-explanatory, with the exception of Spin, Rotate, Swing and Swivel. These 4 animations, around either a vertical axis or a horizontal axis, are available for shapes and images.

Spin is a clockwise or counter-clockwise movement.

Rotate is a 360-degree horizontal or vertical (west to east or north to south) rotation using the vertical or horizontal center of the image as the axis of rotation.

Swing is a 360-degree horizontal or vertical (west to east or north to south) rotation using the right or bottom edge of the image as the axis of rotation.

Swivel is a 180-degree horizontal or vertical (west to east or north to south) rotation and then a 180-degree horizontal or vertical clockwise rotation using the vertical or horizontal center of the image as the axis of rotation.

### d. Continuous ("Loop") Animation

When selected, the animation will continue to repeat itself until any of the following conditions occur.

The Web page is terminated either because it's time setting has been reached or the Web page is left because of an exit event.

The object had been assigned the "Deposit" or "Send Home" movement and a mouse click had occurred.

### C: Child Object Animations and Behaviors

### 1: Application Page Initialization

When an Application Page becomes available all Parent Object Animations immediately begin to execute for that Page. If Child Animations had been defined for the Parent Object, then those animations will begin immediately after the parent object animation concludes. When the child animations conclude, the Child Objects will disappear.

### 2: Parent Object Selected

If a Parent Object is selected in which the "Timeline Activation" event (see chapter 14) has not been selected, then the following would happen. Assuming that this Object has Child Objects which have defined animations, then these animations will begin immediately. The Child Objects will disappear when the Parent Object is no longer selected, or when their animations conclude, whichever occurs first.

### 3: FIRE on Parent Object

If a FIRE event occurs on a Parent Object in which the "Timeline Activation" event (see chapter 14) has been selected, then the following would happen. Assuming that this Object has Child Objects which have defined animations, then these animations will begin immediately. The Child Objects will disappear when their animations conclude.

### D: Defining Object Entry Timelines

Then the Activate Timeline Check Box is selected, all of the Object Entry Animation choices become available.

Settings	Events	Animation	Color 1	Bindings
Activate Timeline D	Object Entry elay (Sec)	Timeline Spec Direction Mo None 💌 No	ifications vemen1 Duration ne - 2 -	on(Sec) Frames .0 - 10 -
Specif Inactive	ications for th	nis Object's En	try Animation A	udio Track Entry Audio File
				Fig 13-

#### 1: Delay

This is the time that will elapse from the moment a Page has become active until the Object appears. This delay can be set to up to 100 seconds.

2: Direction

The way an Object enters the Page can be set. Choose "None" if no direction animation is desired. See Appendix B for the list of the direction entry direction animations available.

3: Movement

The Object's entry to the Page can be extended with movement animations. Choose "None" if no movement animation is desired. See Appendix B for the list of the movement entry direction animations available.

4: Duration

Time is the length of time in seconds that the entry animation will take to complete. It can be set from immediate to up to 100 seconds.

5: Frames (Frames per Second)

The number of frames per second is the number of animations that will occur per second during the entire entry animation process. This can be set from 1 to 50. The higher the number of frames, the smoother the animation will appear.

6: Audio Track

If an audio track has been added to the entry animation, it will start simultaneously with the beginning of the animation. The track can either play once or loop.

Bindings

### E: Defining Object Exit Timelines

Selecting the "Activate" check box will immediately make available all of the Exit Timeline Settings.

Activate

1: Delay

This determines whether the Object will remain on the Page until the Page is unloaded (stay), or whether it should depart after a certain period of time. This time begins from the moment that the object's animation(s) has been completed and all the child timelines, if any, have also been completed. The departure delay time can range up to 100 seconds.

2: Direction

The depart direction is how the Object departs the Page. It can be set to depart immediately (none) or by means of an exit direction. See Appendix B for the list of the exit directions available.

3: Movement

The Object's departure can be enhanced with a Movement anaimation. See Appendix B for the list of 6 exit methods/animations available.

4: Duration

Depart time is the time in seconds that the exit animation will take to complete. It can be set from immediate to up to 100 seconds.

5: Frames (Frames per Second)



Color

Settings Events Animation

**Object Entry Timeline Specifications** 

Fig 13-6

The number of frames is the number of animations that will occur per second during the entire exit animation process. This can be set from 1 to 50.

6: Audio Track

If an audio track has been added to the exit animation, it will start simultaneously with the beginning of the animation. The track can either play once or loop.

### F: Tips

1: Number of frames for button or paragraph zooming animations:

Be sure that the number of frames is less than the point size minus 5 otherwise unpredictable results could occur.

2: Different Animation Effects:

Changing the number of frames to a very small number can dramatically affect the animation effect. Changing the attributes of the object, such as changing the point size and background color for buttons, will create very different visual effects.

3: Animations applied to Animated GIFs:

Animations can also be applied to animated GIFs. Certain animations will work better than others, and the animation effects will be most effective if the timing of the animated GIFs transitions are well synchronized with the speed and the number of frames of the animation. The "Fade In" and "Fade Out" animations will only be effective if the animated GIFs time settings are precisely synchronized with the frames per second of the "fade" animations.

EXPRESS MOBILE User's Guide

, Chapter 14

Working with Events

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 14: Working with Events

A:	Child Object Events	14.04
B:	Exit Events	14.05
C:	Select and Fire Events	14.10
D:	Waypoints	14.13

.

# Chapter 14: Glossary

### A: Child Object Events

#### I: Visibility:

There are three possible techniques for causing child objects to change their visibility.

a. Selection of the Parent Object

Upon selection all child objects will appear (subject to their timeline definitions, if any) and will disappear when the parent object is deselected.

b. Toggle Children on FIRE

When this attribute is selected for the parent object then selection or deselection itself will have no effect. The first FIRE on the parent object the child objects will become visible and remain that way until another FIRE event occurs on that parent object, or when another Parent Object is FIRED on that has the "Hide non-related Children" attribute selected.

c. Child Slide Shows, Timelines and Animations

Child objects can be assigned all the timeline and animation attributes that a parent object can be assigned, and a Child Slide Show can be assigned all the slide transition animations that a Parent Slide Show can be assigned.

The activation of these visual and audio effects begin when the child object becomes visible and ends whenever the child object loses visibility or the timeline is completed, whichever occurs first.

### B: Exit Events

### I: Discussion:

There are currently 28 possible exit events that can be assigned to an Express Mobile Object, which have made the UI very complicated and unwieldy. Although internally the exit events will be stored as per Appendix A, for the UI, the following changes are proposed.

In addition, there are now four new Events:

1: Send IM Alert (Push Rich Content Messaging)

For Sending an IM Alert Message. This can occur by:

- firing on a basic object that has a logical link to a text field or text area
- firing on a List or Dropdown Item .
- firing on a slide with an attached Chat (chat room)

An IM Alert Message will generate, from the Content Server, an IM Alert Page (I may use the System Alert Object if the Alert Page cannot be found.

An IM Alert will have a Text Message defined, which will be extracted from either a designated text field or text area. Optionally, an Avatar, which could be an Image or Video, can be linked to the Alert Page. For Chat, the Avatar will be the Slide Media Object. If the Player is open the IM Alert will immediately Appear as a Modal Alert with Fire enabled for dismissing. If the player is not active, when the player later becomes active there will be an indication of the number of pending IM alerts. The method or methods for browsing the pending alerts is yet to be defined.

#### 2: Set Starting Page

For personalizing the starting page by the consumer. This can occur by firing on a basic object or firing on a List or Dropdown Item. For any subsequent uses of the application, or any "Change View" events in the current session, if the "Change View to Starting Page" is defined, then the consumer-chosen "starting page" will appear.

#### 3: Send String on FIRE

This exit event is available for Text Fields and Text Areas. It will execute the Submit Button Listener so that the current string will be sent to the content Server for processing.

#### 4: Send String on FIRE or Numeric Keys

This exit event is available for Text Fields and Text Areas. It will execute the Submit Button Listener so that the current string will be sent to the content Server for processing.

II: User Interface:

There will be up to six dropdown menus for defining an Event. A subset of these choices will be available, depending upon the type of Object that is selected.

- 1: Alerts:
  - None
  - Display on Fire
  - Display with Synchronization on Fire
  - Display/Hide OnFocus/OffFocus
- 2: Communication
  - None
  - Place Phone Call
  - Send SMS Message
  - Send IM alert
- 3 Share:
  - None
  - Share peer to peer
  - Share multiple selection
  - Share all
- 4 Personalization:
  - Set Add Icon to Icon Strip
  - Remove Icon from Icon Strip
  - Logical First (Starting) Page
  - Set Persistent Variable
- 5 Change View
  - None (next slide)
  - Goto External Page (WAP or HTML)
    - Change View options
    - Replace Current View
    - Launch New Window
    - Replace Top View
  - Goto Internal Page

**Change View Options** 

- Next Internal Page
  - Specific Internal Page
- Specific Internal Page w/Slide Number
- Goto Widget
- Goto Logical First (Starting) Page
- Execute an Xpressmo Application
- Exit Application
- Exit Player
- Execute Scripting Method
  - Pause/Resume Page Timeout
- III: Object Specific Implementations:

As before, events can be assigned to three different object classes. For each class, the following events would be available.

A: All Object Classes and Page Timeout

0=None (or Next Slide)

1=Goto External Page (WAP or HTML)

2=Goto Internal web Page

3=Execute an Xpressmo Application

27=Send IM Alert

B: The Basic Object

4=Place Phone Call

5=Send SMS Message

- 6=Exit Application
- 7=Exit Player

8=Execute Scripting Method

9=Pause/Resume Page Timeout

28=Set Starting Page

29=Set Persistent Variable

C: An Item in a List or Dropdown Object

4=Place Phone Call

5=Send SMS Message

10=	Dropdown:	Add Icon to Icon Strip
	List:	Display/Hide Icon toggle

11= Dropdown: Remove Icon from Icon Strip

28=Set Starting Page

29=Set Persistent Variable

D: A Slide in a Slide Show

12=Goto Specific Slide

E: Web Page

8=Execute Scripting Method

"Goto New View Options" will only be available for the following Exit Events:

1=Goto External Page (WAP/HTML)

2=Goto Internal web Page

"Share Options" will only be available for the following Exit Events:

4=Place Phone Call

5=Send SMS Message

.

### IV: Mapping to Current Player Event Manager:

The Table Below describes what choices will be available by exit event for the six dropdown option menus, and how that maps to the possible list of 28 events.

UI Exit Event	Alert Options	Change View	Personalize	Communicate	Share	Internal
		Options				Exit Event
None	None Display w/Fire Display OnFocus Hide OffFocus Display w/Sync			Send IM Phone Call Phone Call Phone Call Send SMS Send SMS	None Share All None	17 19 22 27 12 24 26 18
				Send SMS	Snare	23
			Addicon	Seliu Sivis	All	13
			Hide Icon			13
			Set Starting			28
			Page			•
Goto External		Replace Current View				1
Page		Launch New Window				2
		Replace Top View				4
Goto Internal	None	Specific/Next Page				3,5
Page	Display w/Sync	Specific/Next Page				15
	None	Logical 1st Page				21
	None	Specific W/Slide #				9
Execute	None					8
Application-	Display					20
Widget	INONE					10
Application						10
Exit Player		+				11
Execute Script		+- <u>-</u>				6
Pause/Resume						7
Page		1				
Goto Slide			•			10
Send IM Alert						27
Send String on FIRF						13
Send String						14
on FIRE or						
Numeric Keys			· ·			
Set Persistent						29
Variable						

#### C: Select and Fire Events

1: Select Events:

The following select events are available.

· Display and activate Children:

If "Toggle Children on FIRE" is not selected, then all children will become visible as soon as the parent object is selected. If any of those children have animations or timelines assigned to them then those animations will begin immediately.

Object Selected Audio:

If an object selected audio track has been assigned to this object then this audio track will immediately start to play. There are four possible settings as shown in the graphic below.



Settings Events Animation Color Bindings **Events and Web Services** None, a series in the series of the series o . Goto External Web Page replacing Current Frame Goto External Web Page Launched in a New Window m Goto a specific Page View Goto External Web Page replacing the Top Frame Goto the next Page View Execute Java Script Method Pause/Resume Page Timeout Execute an Application Goto a Specific Slide in a Page View Exit Application Exit Player Place Phone Call Send String on FIRE Send String on FIRE or Numeric Keys

**Advanced Interactive Settings Mouse State** Timeline Activation Enabled Selected Fire Timeline Entry Suppressed **Object Selected Audio Settings** Enable Server Listener Inactive ☐ Submit Form Toggle Children on FIRE Hide non-related Children Select a Sound File C Object Drag Enabled Fig 14-1

Chapter 14.11

### 2: Fire Events:

The following fire events are available.

- Timeline Activation
- If the "Timeline Activation Enabled" check box is selected, then the parent's animations and timelines will begin upon the detection of a FIRE event. The activation of these animations will not occur simply because the page has become available.
- Timeline Entry Suppressed.
- If the "Timeline Entry Suppressed" check box is selected, then, if there is an entry timeline defined, it will be executed when the page becomes available. It will, however, be suppressed when a FIRE event is detected.
- Display and activate Children:
- If "Toggle Children on FIRE" is selected, then all children will become visible as soon as a FIRE event is
  detected on the parent object. If any of those children have animations or timelines assigned to them then
  those animations will begin immediately. A subsequent FIRE event will cause those children to disappear.
- Hide non-related Children

If the "Hide non-related Children" check box is selected, then any other children that were visible because of the "Display and Activate Children" event described above, then they will immediately disappear.

Object FIRE Audio:

If an object Fire audio track has been assigned to this object then this audio track will immediately start to play. There are four possible settings as shown in the graphic below.

Mouse State
Selected
Fire
Fire Audio Settings



- 3: Tips
  - a: Using Both Timelines and Exit Events:

Because the timeline events will be completed before the exit event for button and image objects, interesting farewell messages can be constructed.

b: Advanced Event based Visual and Audio Effects:

In previous chapters, features and techniques have been discussed which utilized the static and timeline based overlapping of objects with the utilization of various audio and visual attributes that can be specified for these objects. With events, a new universe of user interactions can now be implemented using these dynamic visual and audio interactions between objects, and between the objects' children.

c: Multiple Simultaneous Timeline Events:

You can overlay button and/or image objects on top of each other, each of which may have timeline events associated with them. When the mouse is clicked over these objects all associated timelines will be activated simultaneously.

Chapter 14.13

### D: Waypoints

When an object has an animation direction of either custom or multi-point, waypoints are the starting points, points where there was a pause or direction change and the ending points of an object is course or path. They are identified by blue squares.



### 1: How to Create:

To create waypoints, first select the object to be animated and then choose either the "Custom" or "Multi-Point" animation direction and create your path.

2: How to Define:

To define waypoints, make sure the object is selected and then either choose the "Events" menu and click on "Waypoints" or right click on the object and choose "Waypoints".

Evenis Pag	es Styles
Child Objec	ts §
Waypoints	* ***
Mouse Ever	nts
Animations	
Timelines	



The "Define WayPoints" dialog box will appear. The items in this dialog box tell you what values have been assigned to a particular waypoint.

Def	ine Waypoints for Ar	nimation Trajectorie	S Contraction
Select a Waypoint	Selected Objects	Time (MilliSec)	Leg Length
WayPoint 0		720	148
WayPoint 1		Horizontal	Vertical
WayPoint 2 WayPoint 3		10	10
WayPoint 4		Audio Attached to	Selected Waypoir
	С	Audio Trac	k Pathname
		a and a start of the start of the start of the start of the	
		n Sent Maria ya 2000 ka wa	a prosenta da seriesta. Nomenta da seriesta da serie
	Edit Waynoint	Delete Maynoint	

"Select a Waypoint" shows all the waypoints for the object.

"Selected Objects" shows which object is currently selected for defining, if there are multiple objects on the web page.

"Time" is the length of time the object takes to traverse that leg in the path.

"Horizontal" and "Vertical" shows the current position of that waypoint.

"Leg Length" is the length of the leg beginning with the selected waypoint to the next one.

The "Audio Attached" check box tells you if an audio has been assigned, and, if so, which audio track has been assigned to the selected waypoint.

#### 3: How to Edit:

The "Edit Waypoint" button allows you to change the above positions and lengths. After selecting a waypoint, clicking this button takes you to the "Edit WayPoints" dialog box (below).

The values of the selected waypoint are displayed and can be changed by typing in a new value. An audio track for each waypoint can be changed or added; choose from the Express Mobile audio library or import your own track.

Also any number of objects can be selected for this waypoint. When that waypoint is reached while the object is moving along its trajectory, then any waypoint selected objects, if they have a timeline associated with them, will have their timeline activated.

Edit	the Selected WayPo	int
Selected Waypoint	Horizontal	Vertical
WayPoint 1	158	22
Selected Objects	Object candidates	Leg Length
second base	home	116
	third base second base first base	Time (MilliSec)
	unitaria de la construcción de la c	1993年年1月第一日 1993年年月1日 1993年年年月1日日 1993年年日 1993年年日 1993年年 1993年年 1993年 1997 1997
Audio Attached	Audio Track Pathna	ime
1		

4: How to Delete a WayPoint:

Select the desired waypoint and then touch the "Delete Waypoint" button.

"Audio" tells you which audio track (if any) has been assigned to the selected waypoint.

xpressmo.com

Fig 14-8

.

# EXPRESS MOBILE User's Guide

Chapter 15

Testing Your Application

© 2008 Express Mobile, Inc. All Rights Reserved.

# Chapter 15: Testing Your Application

Testing an Application created in Publisher

15.04

xpressmo.com

1

Chapter 15.3

# Chapter 15 : Glossary

Emulator

Any tool that will, more or less, produce a similar result as what would occur on the Mobile Device itself.

Sun Java Wireless Toolkit for CLDC

WTK

### Testing an Application Created in Publisher

A very convenient way to observer how an application will perform is to select the "Launch" command from the View Menu. As long as one of the supported browsers is available (Opera, Internet Explorer, or FireFox) then the application will be launched in that browser. Note that the Express Mobile Java J2SE Player will be invoked, but the operations will be quite similar, except that the mouse will become the pointing device. If the intention is to run this application on a Java CDC device (such as Set top Boxes, Google Android devices, or Javabased telematic devices, then this will demonstrate almost perfectly how the application will run in these different environments.

Once you have saved an application in Publisher you may wish to see how it would appear on a phone or on a phone emulator. To run the application on a device, the player portion of the application must be downloaded at least once. Once the player is available on the device, you can modify the application, and in most cases, will not need to download the player again. Simply re-launching the application will instantly display the changes.



This discussion assumes a development and test environment. An application

that has been edited may not be immediately available to the end user, as stated here, due to procedural requirements. However, in general, designers and developers should have immediate access to updated applications so that the applications can be tested and validated.

#### **Required Information**

You need the following information before continuing with the player download:

- · The username for Publisher (from the Log In dialog box)
- The server to which Publisher client is connected (from the Log In dialog box)
- The name given to the application (from the Save or Save As dialog box)

### For this example let us use the following names:

- Username: myname
- Server: publisher.com
- Application: MyApp

Address of Application

The web address (also called URL or Uniform Resource Locator) of the mobile application for Java enabled devices has two distinct constructions.

Accessing Applications via Response Director

Response Director provides a single URL for a given application. The Response Director will determine the proper player files to send to the device that is requesting the application.

The construct of a URL for an application is:

http://server/get/username/appname

where:

server is the domain name or IP address of the Publisher server

username is the author's Publisher username

appname is the name given by the author to the mobile application.

For example:

http://publisher.com/get/myname/MyApp

This URL is case sensitive. This URL construct is the preferred method for accessing applications. However, this method may not work for all phones, nor will it work for the phone emulator. If this method does not work, use the alternate method discussed in the following section.
Accessing Applications via Direct URL

This URL construct accesses the .jad file directly. This should be used when testing via the emulator and when the Response Director method does not work for a device. Note that this method may not be available to end users for security reasons. Contact your System Administrator to see if the Direct URL method is available to you.

The Direct URL construct has several options. Initially the most common URL will be described. The format is:

http://server/%7eusername/WebSites/appname/bin/appname.jad

where:

server	is the domain name or IP address of the Publisher server
username	is the application author's Publisher username

appname is the name given by the author to the mobile application.

For example:

http://publisher.com/%7emyname/WebSites/MyApp/bin/MyApp.jad

This URL is case sensitive. This URL is normally for a QVGA sized application with code signing certificate.

All variations of URL are listed in the table below. Note that the QCIF variations may not be available if the author did not explicitly request auto-generation of QCIF version from QVGA version during a Save As operation.

Device Requirements	URL
QVGA Normal	http://server/%7eusemame/WebSites/appname/bin/appname.jad
QVGA No Certificate	http://server/%7eusemame/WebSites/appname/bin/appnameNC.jad
QVGA No File Access	http://server/%7eusername/WebSites/appname/bin/appnameNF.jad
QVGA No Certificate & No File Access	http://server/%7eusername/WebSites/appname/bin/appnameNCNF.jad
BlackBerry	http://server/%7eusername/WebSites/appname/bin/appnameBB.jad
PC Browser	http://server/%7eusername/WebSites/appname/bin/appname.html
QCIF Normal	http://server/%7eusername/WebSites/appnameQCIF/bin/appnameQCIF.jad
QCIF No Certificate	http://server/%7eusername/WebSites/appnameQCIF/bin/appnameQCIFNC.jad
QCIF No File Access	http://server/%7eusername/WebSites/appnameQCIF/bin/appnameQCIFNF.jad
QCIF No Certificate & No File Access	http://server/%7eusername/WebSites/appnameQCIF/bin/appnameQCIFNCNF.jad

Using the Emulator

To use the phone emulator, you must download and install the Sun Java Wireless Toolkit for CLDC (WTK). This is available at:

#### http://java.sun.com/products/sjwtoolkit/

Your PC account must have privileges to install applications. At present, the WTK is only available for Windows and Linux environments.

Once you have installed the WTK, the recommended configuration for the emulator is as follows:

- 1. From Taskbar, go to Start→All Programs→Sun Java (TM) Wireless Toolkit 2.5.2 for CLDC→Preferences.
- 2. Go to the Security Category. For Security Policy, select JTWI; for Security Domain, select maximum.
- 3. Select the OK button and exit the Preferences utility.

Next select the device type that best matches the phone type you want to test. For the most general emulation the recommended setting is as follows:

- 1. From Taskbar, go to Start→All Programs→Sun Java (TM) Wireless Toolkit 2.5.2 for CLDC→Default Device Selection.
- 2. For Default Device, select DefaultColorPhone.
- 3. Select OK to exit the Default Device Selection utility.

To run a mobile Java application in the phone emulator, use your PC browser and type in the URL (web address) of the mobile application using the Direct URL construct. See section **Accessing Applications via Direct URL** for construction of the URL.

#### Publisher Interface

#### Downloading Over the Air (OTA) to a Mobile Device

When available, use the Response Director construct of the URL, as this is the preferred method. (As of this writing, BlackBerry devices must use the Direct URL method to access the player).

Typing in a long URL is tedious on mobile devices with a limited text entry capability. The simplest method is to use SMS to send the URL of the application to the device. For this, you will need the 10-digit NANPA (North American Numbering Plan Area) phone number, or if your SMS service allows, the country code and phone number of your mobile device with which you wish to load and test the application.

First; locate a suitable SMS service. There are several free services available from the PC to send SMS, including Yahoo! and AOL. Not all phones are reachable from some of these free services.

Once the appropriate SMS service has been located, enter in or select the phone number of the mobile device and type the URL of the application into the message area, as described in either section **Accessing Applications via Response Director** or section **Accessing Applications via Direct URL** above.

When your mobile device receives the SMS, open the message and select the URL within the message. Then select the menu option that allows you to link to the URL. These are variously labeled as "Link to" or "Go to" in the device's menu.

Along the way you may encounter some or all of these types of messages, worded differently based on device model:

Type of Message	Appropriate Answer
Confirm download of application MyApp?	Yes
Choice of connection type?	Choose the appropriate highest speed connection.
	The choice will depend on your operator settings.
Application MyApp is untrusted. Download OK?	Yes
Application MyApp version xxx already exists.	Yes
Overwrite with version yyy?	
Remove Previous Data?	If you retain the data, the state of the last run of the application will remain, such as the cache — which will speed up the access. If on the other hand you want to test the pristine state of the application, removal of the application data is recommended.

Table 1 - Types of Messages during Application Player Download

If SMS is not available, you can key in the URL in the mobile device's browser. If you plan to download new versions frequently, you may want to use a bookmarking feature available on most browsers.

.

.

.

# EXPRESS MOBILE Users Guide

Appendix A

Mapping for Direct Text Entry

© 2008 Express Mobile, Inc. All Rights Reserved.

Consecutive presses on the same key without pause allow entry of the following characters.

Number of keypresses on same key without pause $\rightarrow$									
Key↓	1	2	3	4	5	6	7	8	9
0	space	Ó	+						
1	•	,	@	1	?	!	*	#	1
2 abc	а	b	с	2					
3 def	d	e	f	3					
4 ghi	g	h	į	4					
5 jkl	j	k	1	5					
6 mno	m	n	0	6					
7 pqrs	p	q	r	s	7				
8 tuv	t	u	v	8			÷		
9 wxyz	w	x	У	z	9				
*			(rese	rved for la	unch strip	use)		• · · · ·	4
#	# (reserved for launch strip use)								

Pressing one more than the number of defined consecutive keystrokes returns to the first character.

## EXPRESS MOBILE User's Guide

Appendix B

**Object Animations** 

© 2008 Express Mobile, Inc. All Rights Reserved.

**General Directions:** 

1.	Custom	17.	Enter E
2.	Multi-Point	18.	Enter SE
3.	Seek Cursor	19.	Enter S
4.	Attach	20.	Enter SW
5.	Deposit	21.	Enter W
6.	Send Home	22.	Enter NW
7.	Carom N	23.	Exit N
8.	Carom NE	24.	Exit NE
9.	Carom E	25.	Exit E
10.	Carom SE	26.	Exit SE
11.	Carom S	27.	Exit S
12.	Carom SW	28.	Exit SW
13.	Carom W	29.	Exit W
14.	Carom NW	30.	Exit NW
15.	Enter N '	31.	Hover
16.	Enter NE		

General Animations for Text Objects:

.

2. Zoom	11.	
		Grow NW
3. Zoom SE	12.	Grow NE
4. Zoom SW	13.	Grow SE
5. Zoom NW	14.	Grow SW
6. Zoom NE	15.	Shrink SE
7. Fade in	16.	Shrink SW
8. Fade Out	17.	Shrink NW
9. Zoom In	18.	Shrink NE

General Animations for Shapes & Images:

1.	Spin Left		15.	Zoom NW
2.	Spin Right		16.	Zoom NE
3.	Flip Left	for shapes	17.	Fade in
4.	Flip Right	only	18.	Fade Out
5.	Fade		19.	Zoom In
6.	Rotate W-E		20.	Zoom Out
7.	Swing W-E		21.	Grow NW
8.	Swivel W-E		22.	Grow NE
9.	Rotate N-S		23.	Grow SE
10.	Swing N-S		24.	Grow SW
11.	Swivel N-S		25.	Shrink SE
12.	Zoom		26.	Shrink SW
13.	Zoom SE		27.	Shrink NW
14.	Zoom SW		28.	Shrink NE

Entry Motions:

1.	Enter N	5.	Enter S
2.	Enter NE	6.	Enter SW
3.	Enter E	7.	Enter W
4.	Enter SE	8.	Enter NW

#### Entry Methods:

1.	Fade In	4.	Grow NE
2.	Zoom In	5.	Grow SE
3.	Grow NW	6.	Grow SW

Exit Motions:

.

1.	Exit N	5.	Exit S
2.	Exit NE	6.	Exit SW
3.	Exit E	7.	Exit W
4.	Exit SE	8.	Exit NW

#### Exit Methods:

.

.

.

1.	Fade Out	4.	Shrink NE
2.	Zoom Out	5.	Shrink SE
3.	Shrink NW	6.	Shrink SW

### EXPRESS MOBILE User's Guide

Appendix C

Web Component Models

© 2008 Express Mobile, Inc. All Rights Reserved.

#### 1: RSS: USA Today Top Stories

<?xml version="1.0" encoding="utf-8" ?>

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:

schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" name="RSS1" group="Examples" PlayerURL="http://localhost:8080/ wcm/playerreq" ServiceURL="http://rssfeeds.usatoday.com/usatoday-NewsTopStories">

<Inputs />

<Outputs>

<Parameter name="channel-title" dataType="ChannelTitle" parmType="response" ServiceMapping="//rss/channel/title" use="optional">

<Comment>RSS Channel Title</Comment>

</Parameter>

<Parameter name="channel-description" dataType="ChannelDescription" parmType="response"

ServiceMapping="//rss/channel/description" use="optional">

<Comment>RSS Channel Description</Comment>

</Parameter>

<Parameter name="channel-image-url" dataType="ChannelImageURL" parmType="response" ServiceMapping="//rss/channel/image/url" use="optional">

<Comment>RSS Channel Image URL</Comment>

#### </Parameter>

<List ServiceMapping="//rss/channel/item" name="rssitems" controlParm="item-title">

<Parameter name="item-title" data Type="RSSList" parm Type="response"

ServiceMapping="title" use="required">

<Comment>RSS Feed Item title</Comment>

</Parameter>

<Parameter name="item-link" dataType="String" parmType="response"

ServiceMapping="link" use="optional">

<Comment>RSS Feed Item title</Comment>

</Parameter>

<Parameter name="item-description" dataType="multiLineString" parmType="response" ServiceMapping="description" use="optional">

<Comment>RSS Feed Item title</Comment>

</Parameter>

</List>

</Outputs>

</WebComponentModel>

#### 2: Yahoo Search

#### <?xml version="1.0" encoding="utf-8" ?>

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi: schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" name="WebSearch" group="Yahoo" XpathNS="urn:yahoo:srch" PlayerURL="http://localhost:8080/wcm/playerreq" ServiceURL="http://api.search.yahoo.com/Web-SearchService/V1/webSearch?appid=dgSG0s\_V34EXXzvP1a\_evRroETooFkDl9KImJiHL4Ky7hmZ3Of. usRnThe1n.I8-">

#### <Inputs>

- <Parameter name="in1" dataType="String" parmType="request" use="required"
  - ServiceMapping="query">

<Comment>the search term string</Comment>

#### </Parameter>

- <Parameter name="in2" dataType="Int" parmType="request" use="optional" ServiceMapping="results" defaultValue="10">
  - <Comment>The number of results to return, defaults to 10</Comment>

</Parameter>

#### </inputs>

#### <Outputs>

<Parameter ServiceMapping="//ns:ResultSet/@totalResultsAvailable" name="totalresultsAvail" dataType="Int" use="optional">

<Comment>The total results available</Comment>

#### </Parameter>

<Parameter ServiceMapping="//ns:ResultSet/@totalResultsReturned" name="totalresultsReturned" dataType="Int" use="optional">

<Comment>The total results returned</Comment>

#### </Parameter>

<List ServiceMapping="//ns:ResultSet/ns:Result" name="searchResults" controlParm="Title"> <Parameter name="Title" dataType="ComplexList" use="optional" ServiceMapping="Title">

<Comment>The title of the search result link</Comment>

#### </Parameter>

<Parameter name="URL" dataType="String" ServiceMapping="Url" use="optional"> <Comment>The URL to the search result</Comment>

#### </Parameter>

<Parameter name="ClickURL" dataType="String" use="optional" ServiceMapping="ClickUrl"> <Comment>The click thru URL to the search result</Comment>

#### </Parameter>

- <Parameter name="DisplayURL" dataType="String" use="optional" ServiceMapping="DisplayU rl">
- <Comment>A display URL for the user to see for the search result</Comment>

#### </Parameter>

- <Parameter name="Summary" dataType="multiLineString" use="optional" ServiceMapping="S ummary">
  - <Comment>Short paragraph of the search result page</Comment>
- </Parameter>

#### </List>

#### </Outputs>

</WebComponentModel>

#### 3: Yahoo Maps

#### <?xml version="1.0" encoding="utf-8" ?>

- <WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
  - schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" name="Maps" group="Yahoo" PlayerURL="http://localhost:8080/ wcm/playerreq" ServiceURL="http://local.yahooapis.com/MapsService/V1/mapImage?appid=dgSG0s\_ V34EXXzvP1a\_evRroETooFkDl9KImJiHL4Ky7hmZ3Of.usRnThe1n.I8-">

#### <Inputs>

<Parameter name="street" dataType="String" parmType="request" use="optional"

- ServiceMapping="street">
- <Comment>Street Address</Comment>

#### </Parameter>

<Parameter name="city" dataType="String" parmType="request" use="optional"

ServiceMapping="city">

<Comment>City</Comment>

#### </Parameter>

<Parameter name="state" dataType="String" parmType="request" use="optional" ServiceMapping="state">

<Comment>2 Letter state code</Comment>

- </Parameter>
- <Parameter name="zip" dataType="Int" parmType="request" use="optional" ServiceMapping="zip"> <Comment>5 digit zip or 5-4 zip</Comment>
- </Parameter>
- <Parameter name="location" dataType="String" parmType="request" use="optional" ServiceMapping ="location">
  - <Comment>street city state zip location</Comment>
- </Parameter>

<!- need new data type for decimal values † -->

<Parameter name="latitude" dataType="String" parmType="request" use="optional" ServiceMapping= "latitude">

<Comment>float: -90 to 90 The latitude of the starting location.</Comment>

#### </Parameter>

<Parameter name="longitude" dataType="String" parmType="request" use="optional" ServiceMappin g="longitude">

- <Comment>float: -180 to 180 The longitude of the starting location. If both latitude and longitude are specified, they will take priority over all other location data. If only one of latitude or longitude is specified, both will be ignored.</Comment>
- </Parameter>
- <Parameter name="image\_type" dataType="String" parmType="request" use="optional" ServiceMapping="image\_type">
- <Comment>png (default) or gif The image format for the map.</Comment>

</Parameter>

- <Parameter name="image\_height" dataType="Int" parmType="request" use="optional" ServiceMapping="image\_height">
  - <Comment>integer: 10 to 2000 (default: 500) The height of the image being generated, in pixels.</ Comment>

</Parameter>

<Parameter name="image\_width" dataType="Int" parmType="request" use="optional" ServiceMapping="image\_width">

- <Comment>integer: 10 to 2000 (default: 620) The width of the image being generated, in pixels.</ Comment>
- </Parameter>

<Parameter name="zoom" dataType="List" parmType="request" use="optional" ServiceMapping="zoom">

<Comment>integer: 1 to 12 (default: 6) The zoom level for the map, from 1 (street level) to 12

(country level). If a radius is specified, this is ignored.</Comment>

<ListValues value="1">Street Level 1</ListValues>

<ListValues value="2">Street Level 2</ListValues>

<ListValues value="3">Street Level 3</ListValues>

<ListValues value="4">City Level 1</ListValues>

<ListValues value="5">City Level 2</ListValues>

<ListValues value="6">City Level 3</ListValues>

<ListValues value="7">County Level 1</ListValues>

<ListValues value="8">County Level 2</ListValues>

<ListValues value="9">State Level 1</ListValues>

<ListValues value="10">State Level 1</ListValues>

<ListValues value="11">State Level 2</ListValues>

<ListValues value="12">Country Level 1</ListValues>

</Parameter>

<Parameter name="radius" dataType="String" parmType="request" use="optional" ServiceMapping="radius">

<Comment>float How far (in miles) from the specified location to display on the map. The default radius varies according to the location given and the zoom level.</Comment>

</Parameter>

</Inputs>

<Outputs>

<Parameter ServiceMapping="//Result" name="Result" dataType="ImageURL" use="required"> <Comment>The image URL returned</Comment>

</Parameter>

</Outputs>

</WebComponentModel>

#### 4: Movies

<?xml version="1.0" encoding="utf-8" ?>

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:

schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" name="Movies" group="Examples" PlayerURL="http://localhost:8080/ wcm/playerreq" ServiceURL="http://lnx.xpresmo.com/movies\_rss.xml">

<Inputs />

<Outputs>

<Parameter name="channel-title" dataType="ChannelTitle" parmType="response" ServiceMapping="//rss/channel/title" use="optional">

<Comment>RSS Channel Title </Comment>

</Parameter>

<Parameter name="channel-description" dataType="ChannelDescription" parmType="response" ServiceMapping="//rss/channel/description" use="optional">

<Comment>RSS Channel Description </Comment>

</Parameter>

<Parameter name="channel-image-url" dataType="ChannelImageURL" parmType="response" ServiceMapping="//rss/channel/image/url" use="optional">

<Comment>RSS Channel Image URL </Comment>

</Parameter>

<List ServiceMapping="//rss/channel/item" name="rssitems" controlParm="item-title">

<Parameter name="item-title" dataType="RSSList" parmType="response"

ServiceMapping="title" use="required">

<Comment>RSS Item title of video </Comment>

</Parameter>

<Parameter name="item-link" dataType="VideoURL" parmType="response" ServiceMapping="link" use="optional">

<Comment>RSS Item link - RTSP video URL </Comment>

</Parameter>

<Parameter name="item-description" dataType="multiLineString" parmType="response" ServiceMapping="description" use="optional">

<Comment>RSS Item short description of video </Comment>

</Parameter>

</List>

</Outputs>

</WebComponentModel>

#### 5: Accuweather

<?xml version="1.0" encoding="utf-8" ?>

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:

- schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" removeHTML="true" name="Weather2" group="Examples"
- ServiceURL="http://rss.accuweather.com/rss/liveweather\_rss.asp?metric=0">

#### <Managers>

<Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq"/>

<Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq"/>

<Manager type="local" url="http://localhost:8080/wcm/playerreq" />

<Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq" />

<Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq" />

</Managers>

#### <Inputs>

<Parameter ServiceMapping="locCode" name="zip" use="required" dataType="Int"> <Comment>The ZIP code to get weather </Comment>

</Parameter>

#### </Inputs>

#### <Outputs>

<Parameter name="channel-title" dataType="ChannelTitle" parmType="response" ServiceMapping="//rss/channel/title" use="optional">

<Comment>RSS Channel Title </Comment>

#### </Parameter>

<Parameter name="channel-description" dataType="ChannelDescription" parmType="response" ServiceMapping="//rss/channel/description" use="optional">

<Comment>RSS Channel Description </Comment>

#### </Parameter>

<Parameter name="channel-image-url" dataType="ChannelImageURL" parmType="response" ServiceMapping="//rss/channel/image/url" use="optional">

<Comment>RSS Channel Image URL </Comment>

</Parameter>

<!- current weather + -->

<Parameter name="currentweather" dataType="String" parmType="response" ServiceMapping="//rss/ channel/item[1]/title" use="optional">

<Comment>Today's current weather </Comment>

#### </Parameter>

<Parameter name="currentweather-link" dataType="String" parmType="response" ServiceMapping="/ rss/channel/item[1]/link" use="optional">

<Comment>Today's current weather URL </Comment>

#### </Parameter>

<Parameter name="currentweather-description" dataType="multiLineString" parmType="response" ServiceMapping="/rss/channel/item[1]/description" use="optional" containsHTML="true"> <Comment>Today's weather description </Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item[1]/description" name="currentweather-img-src" use="optional" dataType="ImageURL" parmType="EHTML" EmbeddedMapping="img/@src">

<Comment>Today's weather embedded image src of the forecast gif </Comment>

</Parameter>

<!- Today's forecast † -->

<Parameter name="todaysweather" dataType="String" parmType="response" ServiceMapping="//rss/ channel/item[2]/title" use="optional">

<Comment>Today's forecast weather </Comment>

#### </Parameter>

<Parameter name="todaysweather-link" dataType="String" parmType="response" ServiceMapping="/ rss/channel/item[2]/link" use="optional">

<Comment>Today's forecast weather URL </Comment>

#### </Parameter>

- <Parameter name="todaysweather-description" dataType="multiLineString" parmType="response" ServiceMapping="/rss/channel/item[2]/description" use="optional" containsHTML="true"> <Comment>Today's forecast weather description </Comment>
- </Parameter>
- <Parameter ServiceMapping="/rss/channel/item[2]/description" name="todaysweather-img-src" use="optional" dataType="ImageURL" parmType="EHTML" EmbeddedMapping="img/@src">

<Comment>Today's forecast weather embedded image src of the forecast gif </Comment> </Parameter>

<!- Tomorrow's forecast + -->

<Parameter name="tomorrowsweather" dataType="String" parmType="response" ServiceMapping="// rss/channel/item[3]/title" use="optional">

<Comment>Today's forecast weather </Comment>

#### </Parameter>

<Parameter name="tomorrowsweather-link" dataType="String" parmType="response" ServiceMapping="/rss/channel/item[3]/link" use="optional">

<Comment>Today's forecast weather URL </Comment>

#### </Parameter>

<Parameter name="tomorrowsweather-description" dataType="multiLineString"

- parmType="response" ServiceMapping="/rss/channel/item[3]/description" use="optional" containsHTML="true">
- <Comment>Today's forecast weather description </Comment>

#### </Parameter>

<Parameter ServiceMapping="/rss/channel/item[3]/description" name="tomorrowsweather-img-src" use="optional" dataType="ImageURL" parmType="EHTML" EmbeddedMapping="img/@src"> <Comment>Today's forecast weather embedded image src of the forecast gif </Comment> </Parameter>

-

</Outputs>

</WebComponentModel>

#### 6: Yahoo Weather

```
<?xml version="1.0" encoding="utf-8" ?>
<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
```

schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" removeHTML="true" name="YWeather" group="Yahoo" ServiceURL="http://weather.yahooapis.com/forecastrss"> <Managers> <Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq" /> <Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq" /> <Manager type="local" url="http://localhost:8080/wcm/playerreq" /> <Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq" /> <Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq" /> </Managers> <NameSpaces> <NameSpace prefix="yweather" uri="http://xml.weather.yahoo.com/ns/rss/1.0" /> <NameSpace prefix="geo" uri="http://www.w3.org/2003/01/geo/wgs84\_pos#" /> </NameSpaces> <Inputs> <Parameter ServiceMapping="p" name="zip" use="required" dataType="Int"> <Comment>The ZIP code to get weather </Comment> </Parameter> </Inputs> <Outputs> <Parameter name="channel-title" dataType="ChannelTitle" parmType="response" ServiceMapping="//rss/channel/title" use="optional"> <Comment>Channel title with Location selected </Comment> </Parameter> <Parameter name="channel-description" dataType="ChannelDescription" parmType="response" ServiceMapping="//rss/channel/description" use="optional"> <Comment>Channel description with Location selected </Comment> </Parameter> <Parameter name="channel-image-url" dataType="ChannelImageURL" parmType="response" ServiceMapping="//rss/channel/image/url" use="optional"> <Comment>RSS Channel Image URL </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/yweather:location/@city" name="city" use="optional" dataType="String"> <Comment>The Location City </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/yweather:units/@temperature" name="units-temp" use="optional" dataType="String"> <Comment>The units of the temperature , C or F </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/yweather:atmosphere/@humidity" name="atm-humidity" use="optional" dataType="String"> <Comment>The atmosphere humidity </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/yweather:atmosphere/@pressure" name="pressure" use="optional" dataType="String"> <Comment>The atmosphere pressure</Comment> </Parameter>

<Parameter ServiceMapping="/rss/channel/yweather:units/@pressure" name="units-pressure"

use="optional" dataType="String"> <Comment>The units of pressure </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/title" name="title" use="optional" dataType="String"> <Comment>The text "Conditions for (Location) at (Time) </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/geo:lat" name="geo-lat" use="optional" dataType="String"> <Comment>The Latitude of the Location </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/geo:long" name="geo-long" use="optional" dataType="String"> <Comment>The Longitude of the Location </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/link" name="link" use="optional" dataType="String"> <Comment>The URL to the detail page </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/yweather:condition/@text" name="weather-condition-text" use="optional" dataType="String"> <Comment>The short text of the weather condition </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/yweather:condition/@code" name="weather-condition-code" use="optional" dataType="String"> <Comment>The numeric Code of the weather condition </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/yweather:condition/@temp" name="weather-condition-temp" use="optional" dataType="String"> <Comment>The temperature </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/yweather:condition/@date" name="weather-condition-date" use="optional" dataType="String"> <Comment>The date </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/description" name="weather-description" use="optional" containsHTML="true" dataType="multiLineString"> <Comment>The HTML version of the weather condition </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/yweather:forecast[1]/@day" name="today-forecastday" use="optional" dataType="String"> <Comment>Today's day of week </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/yweather:forecast[1]/@date" name="today-forecastdate" use="optional" dataType="String"> <Comment>Today's date </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/yweather:forecast[1]/@low" name="today-forecastlow" use="optional" dataType="String"> <Comment>Today's forecast Low </Comment> </Parameter> <Parameter ServiceMapping="/rss/channel/item/yweather:forecast[1]/@high" name="today-forecasthigh" use="optional" dataType="String">

<Comment>Today's forecast High </Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item/yweather:forecast[1]/@text" name="today-forecasttext" use="optional" dataType="String">

<Comment>Today's forecast text </Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item/yweather:forecast[1]/@code" name="today-forecastcode" use="optional" dataType="String">

<Comment>Today's forecast code </Comment>

</Parameter>

<!- tomorrow t -->

<Parameter ServiceMapping="/rss/channel/item/yweather:forecast[2]/@day" name="tomorrow-forecast-day" use="optional" dataType="String">

<Comment>Tomorrow's day of week </Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item/yweather:forecast[2]/@date" name="tomorrow-forecast-date" use="optional" dataType="String">

<Comment>Tomorrow's date </Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item/yweather:forecast[2]/@low" name="tomorrow-forecast-low" use="optional" dataType="String">

<Comment>Tomorrow's forecast Low </Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item/yweather:forecast[2]/@high" name="tomorrow-forecast-high" use="optional" dataType="String">

<Comment>Tomorrow's forecast High </Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item/yweather.forecast[2]/@text" name="tomorrow-forecast-text" use="optional" dataType="String">

<Comment>Tomorrow's forecast text </Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item/yweather:forecast[2]/@code" name="tomorrow-forecast-code" use="optional" dataType="String">

<Comment>Tomorrow's forecast code </Comment>

</Parameter>

</Outputs>

</WebComponentModel>

Appendix C.12

#### 7: RSS Stock Quotes

<?xml version="1.0" encoding="utf-8" ?>

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi: schemaLocation="http://webcomponents.xpressmo.com ./webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" removeHTML="true" name="Stocks" group="Examples" ServiceURL="http://www.quoterss.com/quote.php?frmt=0&Freq=0"> <Managers> <Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq" /> <Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq" /> <Manager type="local" url="http://localhost:8080/wcm/playerreq" /> <Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq" /> <Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq" /> </Managers> <Inputs> <Parameter ServiceMapping="symbol" name="symbol" use="required" dataType="String"> <Comment>The stock symbol code to get quote </Comment> </Parameter> </lnputs> <Outputs> <Parameter name="channel-title" dataType="ChannelTitle" parmType="response" ServiceMapping="//rss/channel/title" use="optional"> <Comment>RSS Channel Title : symbol requested </Comment> </Parameter> <Parameter name="channel-description" dataType="ChannelDescription" parmType="response" ServiceMapping="//rss/channel/description" use="optional"> <Comment>RSS Channel Description : quote RSS attribution </Comment> </Parameter> <List ServiceMapping="//rss/channel/item" name="rssitems" controlParm="item-title"> <Parameter name="item-title" dataType="RSSList" parmType="response" ServiceMapping="title" use="required"> <Comment>Main quote string </Comment> </Parameter> <Parameter name="item-link" dataType="String" parmType="response" ServiceMapping="link" use="optional"> <Comment>quote url </Comment> </Parameter> <Parameter name="item-description" dataType="multiLineString" parmType="response" ServiceMapping="description" use="optional" containsHTML="true"> <Comment>quote description contains HTML </Comment> </Parameter> <Parameter ServiceMapping="description" name="change" use="optional" dataType="String" parmType="EHTML" EmbeddedMapping="table/tbody/tr[1]/td[1]/table[1]/tbody[1]/tr[3]/ td[2]/font"> <Comment>The change for this stock, embedded in HTML </Comment> </Parameter> <Parameter ServiceMapping="description" name="percentChange" use="optional" dataType="String" parmType="EHTML" EmbeddedMapping="table/tbody/tr[1]/td[1]/table[1]/ tbody[1]/tr[4]/td[2]/font"> <Comment>The percent change for this stock, embedded in HTML </Comment> </Parameter> </List> </Outputs>

#### 8: Map Quest

<?xml version="1.0" encoding="utf-8"?>

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi: schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" ServiceURL="action" group="AOL" name="MapQuestCluster"> <Managers>

#### cmanagers>

<Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq" />

<Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq" />

<Manager type="local" url="http://localhost:8080/wcm/playerreq" />

<Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq" />

<Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq" />

#### </Managers>

#### <Actions>

<Action name="panN" class="com.xpressmo.webcomponents.mapquest.MapClusterAction" />
<Action name="panS" class="com.xpressmo.webcomponents.mapquest.MapClusterAction" />
<Action name="panE" class="com.xpressmo.webcomponents.mapquest.MapClusterAction" />
<Action name="panW" class="com.xpressmo.webcomponents.mapquest.MapClusterAction" />
<Action name="getPOI" class="com.xpressmo.webcomponents.mapquest.MapClusterAction" />
</Action name="getMapCluster" class="com.xpressmo.webcomponents.mapquest.MapCluster" class="com.xpressmo.webcomponents.mapquest.MapCluster" class="com.xpressmo.webcomponents.mapquest.MapCluster" class="com.xpressmo.webcomponents.mapquest.MapCluster" class="com.x

#### </Actions>

#### <Inputs>

<Parameter ServiceMapping="TXID" name="TXID" dataType="String" use="optional">

<Comment>A transaction ID that the player uses when it needs to batch requests in session (needs new dataType possibly)</Comment>

#### </Parameter>

<Parameter ServiceMapping="zip" name="zip" dataType="String" use="required">

<Comment>Zip code for map </Comment>

#### </Parameter>

<Parameter ServiceMapping="street" name="street" dataType="String" use="optional"> <Comment>Street address for map </Comment>

</Parameter>

<Parameter ServiceMapping="POI" name="POI" dataType="MultiSelectList" use="optional" multiselectDelimiter="|">

<Comment>Points of interest to use for the getPOI action </Comment>

<ListValues value="3014">GAS STATION </ListValues>

<ListValues value="3016">RESTAURANT </ListValues>

<ListValues value="3021">HOTEL/GUEST HOUSE </ListValues>

<ListValues value="3025">PARKING LOT </ListValues>

<ListValues value="3029">CINEMA </ListValues>

<ListValues value="3002">ATM </ListValues>

<ListValues value="3010">AIRPORT </ListValues>

<ListValues value="3005">BUS STATION </ListValues>

<ListValues value="3003">TRAIN STATION </ListValues>

</Parameter>

<Parameter ServiceMapping="height" name="height" dataType="Int" use="required"> <Comment>The returned image height </Comment>

</Parameter>

<Parameter ServiceMapping="width" name="width" dataType="Int" use="required"> <Comment>The returned image width </Comment>

</Parameter>

<Parameter ServiceMapping="mapScale" name="mapScale" dataType="PipelineMultipleSelect"

use="required" defaultValue="48000" multiselectDelimiter="|">

<Comment>Map scale is defined as the proportion of map units to real-world units. For example, if a mapis scale is expressed as 10,000, this means a scale ratio of 1:10,000, meaning one map unit represents 10,000 realworld units. On a map where one inch equals 5 miles, the scale ratio is 1:316,800 because there are 316,800 inches in 5 miles. A typical city-level map might have a scale of 1:24,000, while a typical state-level map might have a scale of 1:2,000,000. </Comment>

<ListValues value="6000">Street 1 </ListValues> <ListValues value="12000">Street 2 </ListValues> <ListValues value="24000">City 1 </ListValues> <ListValues value="48000">City 2 </ListValues> <ListValues value="96000">City 3 </ListValues> <ListValues value="192000">County 1 </ListValues> <ListValues value="400000">County 2 </ListValues> <ListValues value="400000">County 2 </ListValues> <ListValues value="600000">County 3 </ListValues> <ListValues value="1600000">State 1 </ListValues> <ListValues value="300000">State 2 </ListValues> <ListValues value="1600000">State 3 </ListValues> <ListValues value="6000000">Country 1 </ListValues> <ListValues value="12000000">Country 1 </ListValues> <ListValues value="12000000">Country 2 </ListValues>

<ListValues value="48000000">Continent </ListValues>

</Parameter>

<Parameter ServiceMapping="Lat" name="Lat" dataType="String" use="optional"> <Comment>The latitude to use instead of address </Comment>

</Parameter>

<Parameter ServiceMapping="Lng" name="Lng" dataType="String" use="optional"> <Comment>The longitude to use instead of address </Comment>

</Parameter>

</lnputs>

<Outputs>

<Parameter ServiceMapping="Lat" name="Lat" dataType="String" use="optional"> <Comment>The latitude returned from the getmap </Comment>

</Parameter>

<Parameter ServiceMapping="Lng" name="Lng" dataType="String" use="optional"> <Comment>The longitude returned from the getmap </Comment>

</Parameter>

<Parameter ServiceMapping="POI-MapImage" name="POI-MapImage" dataType="ImageData" use="optional">

<Comment>The image data base 64 encoded for a POI map </Comment> </Parameter>

VI ai ai ai ai cici.>

<Parameter ServiceMapping="POI-MapURL" name="POI-MapURL" dataType="ImageURL" use="required">

<Comment>The output map URL for a POI map </Comment>

</Parameter>

<List ServiceMapping="mapList" name="mapList">

<!-- for clustering requests and outputs to reduce net usage--->

<Parameter ServiceMapping="mapurl" name="mapurl" dataType="ImageURL" use="required">

<Comment>The output map URL </Comment>

</Parameter>

<Parameter ServiceMapping="mapScaleforURL" name="mapScaleforURL" dataType="String" use="required">

<Comment>This is the mapscale that the corresponding map URL is for </Comment> </Parameter>

- <Parameter ServiceMapping="mapImage" name="mapImage" dataType="ImageListData" use="optional">
- <Comment>The image data base 64 encoded </Comment>

</Parameter>

#### </List>

- <List ServiceMapping="POIdataList" name="POIdataList" controlParm="name">
  - <Parameter ServiceMapping="pointName" name="pointName" dataType="ComplexList" use="optional">
    - <Comment>The point of interest Name </Comment>
  - </Parameter>
  - <Parameter ServiceMapping="pointKey" name="pointKey" dataType="String" use="optional"> <Comment>The point of interest Key </Comment>
  - </Parameter>
  - <Parameter ServiceMapping="pointDistance" name="pointDistance" dataType="String" use="optional">
    - <Comment>The point of interest distance in miles </Comment>
  - </Parameter>
  - <Parameter ServiceMapping="pointAddress" name="pointAddress" dataType="String" use="optional">
    - <Comment>The street address for the point of interest </Comment>
  - </Parameter>
  - <Parameter ServiceMapping="pointPhoneNumber" name="pointPhoneNumber" dataType="String" use="optional">
    - <Comment>The phone number for the point of interest </Comment>
  - </Parameter>
- </List>
- </Outputs>

</WebComponentModel>

#### 9: ThumbPlay Storefront

<?xml version="1.0" encoding="utf-8"?>

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi: schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" ServiceURL="http://api.hosted.thumbplay.com/api/rest/search/conte nt?pname=1244&maxReturnedResults=10" group="ThumbPlay" name="Search"> <Managers> <Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq" /> <Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq" /> <Manager type="local" url="http://localhost:8080/wcm/playerreg" /> <Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq" /> <Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq" /> </Managers> <NameSpaces> <NameSpace prefix="ns2" uri="http://search.service.api.thumbplay.com/" /> </NameSpaces> <Inputs> <Parameter ServiceMapping="keywords" name="keywords" dataType="String" use="required"> <Comment>The search keywords </Comment> </Parameter> <Parameter ServiceMapping="contentTypes" name="contentTypes" dataType="MultiSelectList" use="required" multiselectDelimiter="|"> <Comment>Pipe (iji) separated values, One of: Music ringtone, Ringtone, Wallpaper, Animation, Game, Video </Comment> <ListValues value="Music ringtone">Music ringtone </ListValues> <ListValues value="Ringtone">Ringtone </ListValues> <ListValues value="Wallpaper">Wallpaper </ListValues> <ListValues value="Animation">Animation </ListValues> <ListValues value="Game">Game </ListValues> <ListValues value="Video">Video </ListValues> </Parameter> </Inputs> <Outputs> <List ServiceMapping="//ns2:findContentResponse/contentResults/contentResult"</pre> name="resultList" controlParm="title"> <Parameter ServiceMapping="albumName" name="albumName" dataType="String" use="optional"> <Comment>The album name </Comment> </Parameter> <Parameter ServiceMapping="artist" name="artist" dataType="String" use="optional"> <Comment>The artist name </Comment> </Parameter> <Parameter ServiceMapping="artistId" name="artistId" dataType="String" use="optional"> <Comment>The artist ID </Comment> </Parameter> <Parameter ServiceMapping="contentId" name="contentId" dataType="String" use="optional"> <Comment>The content ID </Comment> </Parameter> <Parameter ServiceMapping="contentType" name="contentType" dataType="String" use="optional"> <Comment>The content type </Comment> </Parameter>

<Parameter ServiceMapping="imageSmallURL" name="imageSmallURL" dataType="ImageURL" use="optional">

<Comment>The small image URL 75x75 jpg </Comment>

</Parameter>

- <Parameter ServiceMapping="imageMediumURL" name="imageMediumURL" dataType="ImageURL" use="optional">
  - <Comment>The medium image URL 150x150 jpg </Comment>

</Parameter>

- <Parameter ServiceMapping="mobilePurchaseURL" name="mobilePurchaseURL" dataType="PurchaseURL" use="optional">
  - <Comment>The mobile (WAP) purchase page URL </Comment>

</Parameter>

- <Parameter ServiceMapping="previewURL" name="previewURL" dataType="AudioURL" use="optional">
  - <Comment>The mp3 preview URL, if ringtones found </Comment>

</Parameter>

<Parameter ServiceMapping="title" name="title" dataType="ComplexList" use="optional"> <Comment>The content title </Comment>

</Parameter>

</List>

#### </Outputs>

</WebComponentModel>

#### 10: Facebook

<?xml version="1.0" encoding="utf-8"?>

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi: schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http:// webcomponents.xpressmo.com" ServiceURL="action" group="FaceBook" name="FBActions"> <Managers> <Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq" /> <Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq" /> <Manager type="local" url="http://localhost:8080/wcm/playerreq" /> <Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq" /> <Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq" /> </Managers> <Actions> <Action name="getFriends" class="com.xpressmo.webcomponents.facebook.FaceBookAction" /> <Action name="setStatus" class="com.xpressmo.webcomponents.facebook.FaceBookAction" /> <Action name="sendIM" class="com.xpressmo.webcomponents.facebook.FaceBookAction"/> </Actions> <Inputs> <Parameter ServiceMapping="fbsession" name="fbsession" dataType="String" use="required"> <Comment>The infinite session key from FaceBook for our application for the test user </Comment> </Parameter> <Parameter ServiceMapping="IM" name="IM" dataType="String" use="optional"> <Comment>The instant message text to send for sendIM </Comment>

</Parameter>

<Parameter ServiceMapping="uid" name="uid" dataType="Int" use="optional">

<Comment>The FaceBook user id to send a message to for sendIM </Comment> </Parameter>

<Parameter ServiceMapping="statusMessage" name="statusMessage" dataType="String" use="optional">

<Comment>The status message text to set for setStatus </Comment>

</Parameter>

</Inputs>

<Outputs>

<List ServiceMapping="friendList" name="friendList" controlParm="uid">

<Parameter ServiceMapping="uid" name="uid" dataType="Int" use="optional"> <Comment>The FaceBook uid for that friend </Comment>

</Parameter>

<Parameter ServiceMapping="name" name="name" dataType="String" use="optional"> <Comment>The friend's name </Comment>

</Parameter>

<Parameter ServiceMapping="pic" name="pic" dataType="ImageURL" use="optional"> <Comment>The URL to the friend's thumbnail JPG </Comment>

</Parameter>

<Parameter ServiceMapping="status" name="status" dataType="String" use="optional"> <Comment>The friend's status message, may be blank </Comment>

</Parameter>

</List>

</Outputs>

</WebComponentModel>

### How to create a Web Component Model

#### REST XML Example

Get the schema XSD webcomponentmodel.xsd to validate your model XML from the installation at Publishing-Server/xsd.

Load into Eclipse or Oxygen IDE

Create a new XML file and associate the schema

Add the name, group schema attributes and namespace

You should have an example of the feed XML that is returned also to construct the XPATH mappings

Setup a model for the USA Today news RSS feed. Add the Feed URL as the Service URL, add the Managers and Manager Elements as the URLs to the web component manager you are using.

«WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http://webcomponents.xpressmo.com" name="RSS1" group="Examples"

ServiceURL="http://rssfeeds.usatoday.com/usatoday-NewsTopStories" >

<Managers>

<Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq"/> <Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq"/>

<Manager type="local" url="http://localhost:8080/wcm/playerreq"/>

· <Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq"/>

<Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq"/></Managers>

</WebComponentModel>

Here is a USA Today news RSS feed example URL http://rssfeeds.usatoday.com/usatoday-NewsTopStories

Define the inputs to the service For this service there are no inputs, it is just a RSS feed.

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http://webcomponents.xpressmo.com" name="RSS1" group="Examples" ServiceURL="http://rssfeeds.usatoday.com/usatoday-NewsTopStories"> <Manage:s>

<Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq"/> <Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq"/>

Appendix C.20

Publisher Interface

«Manager type="local" url="http://localhost:8080/wcm/playerreq"/>

<Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq"/>

<Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq"/>

</Managers>

<Inputs/>

</WebComponentModel>

Define the outputs from the service

We want 3 parameters mapped to channel data, channel-title, channel-description, channel-image-url. Map the Elements needed with XPATH in the ServiceMapping attribute. Add the dataTypes to hint the GUI needed for the Author. Add comments and use.

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http://webcomponents.xpressmo.com" name="RSS1"

group="Examples"

ServiceURL="http://rssfeeds.usatoday.com/usatoday-NewsTopStories" > <Managers>

<Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq"/>

<Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq"/>

<Manager type="local" url="http://localhost:8080/wcm/playerreq"/>

<Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq"/>

<Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq"/>

</Managers>

<Inputs/>

<Outputs>

<Parameter name="channel-title" dataType="ChannelTitle" ServiceMapping="//rss/channel/title" use="optional" > <Comment>RSS Channel Title</Comment>

«/Parameter>

<Parameter name="channel-description" dataType="ChannelDescription" ServiceMapping="//rss/channel/ description" use="optional" >

<Comment>RSS Channel Description</Comment>

</Parameter>

<Parameter name="channel-image-url" dataType="ChannelImageURL" ServiceMapping="//rss/channel/image/url" use="optional" >

<Comment>RSS Channel Image URL</Comment>

«/Parameter>

«/Outputs>

</WebComponentModel>

Define any List outputs from the service

Add the item list of parameters to the model. We want item-title, item-link, item-description.

<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http://webcomponents.xpressmo.com" name="RSS1"

group="Examples"

ServiceURL="http://rssfeeds.usatoday.com/usatoday-NewsTopStories" > <Managers>

<Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq"/> <Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq"/>

<Manager type="local" url="http://localhost:8080/wcm/playerreq"/>

<Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq"/>

<Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq"/>

</Managers>

<Inputs/>

#### <Outputs>

<Parameter name="channel-title" dataType="ChannelTitle" ServiceMapping="//rss/channel/title" use="optional" > <Comment>RSS Channel Title</Comment>

</Parameter>

<Parameter name="channel-description" dataType="ChannelDescription" ServiceMapping="//rss/channel/
description" use="optional" >

<Comment>RSS Channel Description</Comment>

</Parameter>

<Parameter name="channel-image-url" dataType="ChannelImageURL" ServiceMapping="//rss/channel/image/url" use="optional" >

<Comment>RSS Channel Image URL</Comment>

</Parameter>

<List ServiceMapping="//rss/channel/item" name="rssitems" controlParm="item-title">

<Parameter name="item-title" dataType="RSSList"

ServiceMapping="title" use="required" >

<Comment>RSS Feed Item title</Comment>

</Parameter>

<Parameter name="item-link" dataType="String"

ServiceMapping="link" use="optional" >

<Comment>RSS Feed Item link</Comment>

</Parameter>

<Parameter name="item-description" dataType="multiLineString"

ServiceMapping="description" use="optional" containsHTML="true">

<Comment>RSS Feed Item description</Comment>

</Parameter>

```
</List>
```

</Outputs>

«/WebComponentModel>

Define any Namespaces used by the service None used here, we can omit this Element.

Load the model into the web component manager webapp in Tomcat. Copy to %CATALINA\_HOME%\webapps\wcm\webcomponents directory and reload the Registry. Go to http://<wcm host>/wcm/console.htm Registry tab and press "reload registry".

Load the model into the PublisherServer/Libraries/webcomponents directory for use by the Author. Copy to %PUBLISHER\_HOME\Libraries\webcomponents.

Test the model using the console at http://<wcm host>/wcm/console.htm where <wcm host> is where you are running the manager.

For the input PlayerMessage JSON string we use:

{"class":"class com xpressmo webcomponents service.PlayerMessage",

#### "inputs":{],

"webcomponentGroup":"Examples",

"webcomponentName":"RSS1"}

You should get a JSON response back similar to:

["cacheKey":"Examples\RSS1","inputs":{},"outputList":[{"listMap":{"item-description":"All the children taken from a polygamist group's Texas ranch are now back with their parents. ","itemlink":"http:\/\rssfeeds.usatoday.com\/~r\/usatoday-NewsTopStories\/~3\/304500298\/200 8-06-04-polygamists-children\_N.htm","item-title":"Texas: All sect kids returned"},"listName" :"rssitems"},{"listMap":{"item-description":"One person was killed as a line of severe weather moved through the Washington area on Wednesday, downing power lines and trees ... ","itemlink":"http:\/\rssfeeds.usatoday.com\/~r\/usatoday-NewsTopStories\/~3\/304835640\/2008-06-04-washington-storms\_N.htm","item-title":"1 killed as severe storms

url":"http:///images.usatoday.com/\_common/\_images/usatoday.com\_135x20.gif"},"responseCo de":0,"responseString":"","webcomponentAction":"","webcomponentGroup":"Examples","webco mponentName":"RSS1"}

#### More Advanced Model

Now let's look at a more advanced model: Yahoo Search

<?xml version="1.0" encoding="utf-8"?>

<WebComponentModel

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http://webcomponents.xpressmo.com" name="WebSearch" group="Yahoo" ServiceURL="http://api.search.yahoo.com/WebSearchService/V1/ webSearch?appid=<appidxyz>">

<Managers>

<Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq"/>

<Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq"/>

<Manager type="local" url="http://localhost:8080/wcm/playerreq"/>

<Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq"/>

<Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq"/>

</Managers>

«NameSpaces»

<NameSpace prefix="yh" uri="urn:yahoo:srch"/> </NameSpaces>

<Inputs>

<Parameter name="in1" dataType="String" parmType="request" use="required" ServiceMapping="query">

<Comment>the search term string</Comment>

</Parameter>

<Parameter name="in2" dataType="Int" parmType="request" use="optional" ServiceMapping="results" defaultValue="10">

<Comment>The number of results to return, defaults to 10</Comment>

</Parameter>

</hputs>

<Outputs>

- <Parameter ServiceMapping="//yh:ResultSet/@totalResultsAvailable" name="totalresultsAvail" dataType="Int" use="optional">
- <Comment>The total results available</Comment>

</Parameter>

- <Parameter ServiceMapping="//yh:ResultSet/@totalResultsReturned"
- name="totalresultsReturned" dataType="Int" use="optional">
- <Comment>The total results returned</Comment>
- </Parameter>

<List ServiceMapping="//yh:ResultSet/yh:Result" name="searchResults" controlParm="Title">

<Parameter name="Title" dataType="ComplexList" use="optional" ServiceMapping="yh:Title"> <Comment>The title of the search result link</Comment>

</Parameter>

<Parameter name="URL" dataType="URL" ServiceMapping="yh:Url" use="optional">

- <Comment>The URL to the search result</Comment>
- </Parameter>

<Parameter name="ClickURL" dataType="URL" use="optional" ServiceMapping="yh:ClickUrl"> <Comment>The click thru URL to the search result</Comment>

</Parameter>

<Parameter name="DisplayURL" dataType="URL" use="optional"

ServiceMapping="yh:DisplayUrl">

<Comment>A display URL for the user to see for the search result</Comment> </Parameter>

- <Parameter name="Summary" dataType="multiLineString" use="optional"
- ServiceMapping="yh:Summary">

<Comment>Short paragraph of the search result page</Comment>

- </Parameter>
- </list>
- </Outputs>
- </WebComponentModel>

The hi-lighted areas show what is new for this model. The namespace that Yahoo uses is defined, yh prefix is selected to tie to the URI, and used in the ServiceMapping for Output Parameters. This service uses 2 inputs, in1 is a search string and mapped to the query parameter, in2 is a number to specify how many results to return, with a default of 10. These are put into the URL by the WebComponentManager when parsing the inputs and requesting the Service URL.

Model with Embedded HTML returned from Service

This model uses Accuweather RSS feeds which return embedded HTML within the item list. In an excerpt from the Accuweather RSS feed, the description element contains an embedded image tag.

<description>Currently in PLEASANT HILL, CA: 70 &#176;F and Partly Sunny &lt;img src="http:// vortex.accuweather.com/phoenix2/images/common/icons/02\_31x31.gif"> </description>

The model for this Webcomponent: the highlighted areas are the new features used for this model.

<?xml version="1.0" encoding="utf-8"?>

<WebComponentModel xrnlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://webcomponents.xpressmo.com ./webcomponentmodel.xsd" xmlns="http://webcomponents.xpressmo.com" removeHTML="true" name="Weather2" group="Examples"

ServiceURL="http://rss.accuweather.com/rss/liveweather\_rss.asp?metric=0">

#### <Managers>

<Manager type="qa" url="http://qa.xpresmo.com/wcm/playerreq"/> <Manager type="dev" url="http://dev.xpresmo.com/wcm/playerreq"/> <Manager type="local" url="http://localhost:3080/wcm/playerreq"/> <Manager type="staging" url="http://stg.xpresmo.com/wcm/playerreq"/> <Manager type="product" url="http://cst.xpresmo.com/wcm/playerreq"/> </Managers>

<Inputs>

<Parameter ServiceMapping="locCode" name="zip" use="required" dataType="Int" defaultValue="94949">

<Comment>The ZIP code to get weather, default is 94949 </Comment> </Parameter>

</inputs>

<Outputs>

<Parameter name="channel-title" dataType="ChannelTitle" parmType="response" ServiceMapping="//rss/channel/title" use="optional"> <Coroment>RSS Channel Title</Comment>

</Parameter>

<Parameter name="channel-description" dataType="ChannelDescription" parmType="response" ServiceMapping="//rss/channel/description" use="optional">

<Comment>RSS Channel Description</Comment>

</Parameter>

<Parameter name="channel-image-url" dataType="ChannelImageURL" parmType="response" ServiceMapping="//rss/channel/image/url" use="optional">

<Comment>RSS Channel Image URL</Comment>

</Parameter>

<!-- current weather -->

<Parameter name="currentweather" dataType="String" parmType="response" ServiceMapping="//rss/channel/item[1]/title" use="optional">

<Comment>Today's current weather</Comment>

</Parameter>

<Parameter name="currentweather-link" dataType="String" parmType="response" ServiceMapping="/rss/channel/item[1]/link" use="optional"> <Comment>Today's current weather URL</Comment>

</Parameter>

<Parameter name="currentweather-description" dataType="multiLineString" parmType="response"

ServiceMapping="/rss/channel/item[1]/description" use="optional" containsHTML="true"> <Comment>Today's weather description</Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item[1]/description" name="currentweather-img-src" use="optional" dataType="ImageURL" parmType="EHTML" EmbeddedMapping="img/@src"> <Comment> Today's weather embedded image <u>src</u> of the forecast <u>gif</u></Comment> </Parameter>

<!-- Today's forecast -->

<Parameter name="todaysweather" dataType="String" parmType="response" ServiceMapping="//rss/channel/item[2]/title" use="optional">
<Comment>Today's forecast weather</Comment>

</Parameter>

<Parameter name="todaysweather-link" dataType="String" parmType="response" ServiceMapping="/rss/channel/item[2]/link" use="optional">

<Comment>Today's forecast weather URL</Comment>

</Parameter>

<Parameter name="todaysweather-description" dataType="multiLineString" parmType="response"

ServiceMapping="/rss/channel/item[2]/description" use="optional" containsHTML="true"> <Comment>Today's forecast weather description</Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item[2]/description" name="todaysweather-img-src" use="optional" dataType="ImageURL" parmType="EHTML" EmbeddedMapping="img/@src">

<Comment>Today's forecast weather embedded image src of the forecast gif</Comment> </Parameter>

<!-- Tomorrow's forecast -->

<Parameter name="tomorrowsweather" dataType="String" parmType="response" ServiceMapping="//rss/channel/item[3]/title" use="optional">

<Comment>Today's forecast weather</Comment>

</Parameter>

<Parameter name="tomorrowsweather-link" dataType="String" parmType="response" ServiceMapping="/rss/channel/item[3]/link" use="optional">

<Comment>Today's forecast weather URL</Comment>

</Parameter>

<Parameter name="tomorrowsweather-description" dataType="multiLineString" parmType="response" ServiceMapping="/rss/channel/item[3]/description" use="optional" containsHTML="true">

<Comment>Today's forecast weather description</Comment>

</Parameter>

<Parameter ServiceMapping="/rss/channel/item[3]/description" name="tomorrowsweather-imgsrc"

use="optional" dataType="ImageURL" parmType="EHTML" EmbeddedMapping="img/@src"> <Comment>Today's forecast weather embedded image src of the forecast gif</Comment> </Parameter>

#### </Outputs>

</WebComponentModel>

The removeHTML=true attribute turns on the embedded HTML removal feature of WCM. This will strip out any HTML tags, leaving only the text for each Parameter marked with containsHTML="true". The Parameter currentweather-description has this feature enabled.

Parmtype="EHTML" is used to tell WCM to look into the embedded HTML as a XML fragment and pull out the Element or Attribute you are interested in. The Parameter currentweather-img-src uses this feature to extract the embedded img tag src attribute. The EmbeddedMapping="img/@src" is an XPATH expression into the embedded XML fragment. The Parameter's ServiceMapping="/rss/channel/item[1]/description" attribute is the XPATH expression into the main XML document.

xpressmo.com

xpressmo.com

IPR2021-01146 Page 00506

EXPRESS MOBILE User's Guide Appendix D Landable

© 2008 Express Mobile, Inc. All Rights Reserved.

- 1. All Parent Video Objects
- 2. All Parent Slide Show Objects
- 3. GIS Object (N/A)
- 4. Virtual Tour Object(N/A)
- 5. All Form Objects with the following exceptions:
  - a. Text Field with non-Selectable attribute (ObjectBooleanAttribute[3]=false)
  - b. Text Area with non-Selectable attribute (ObjectBooleanAttribute[3]=false)
- 6. All Parent Objects that have one or more Child Objects
- 7. All Parent Objects with an exit Event (ObjectIntegerAttribute[12] > 0)
- 8. All Parent Objects with a Selection Cursor Defined (ObjectIntegerAttribute[14]>0)
- 9. All Parent Objects with a FIRE Cursor Defined (ObjectIntegerAttribute[15]>0)
- 10. All Parent Objects with a Selection Audio Defined (ObjectIntegerAttribute[16]>0)
- 11. All Parent Objects with a FIRE Audio Defined (ObjectIntegerAttribute[17]>0)
- 12. All Parent Objects with TimeLine Events (ObjectBooleanAttribute[6])
- 13. All Parent Objects with Draggable Attribute (ObjectBooleanAttribute[8])
- 14. All Parent Objects with Submit Form Attribute (ObjectBooleanAttribute[16])
- 15. All Objects that have a web component attached to them

xpressmo.com

xpressmo.com

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US2009/039695

International filing date: 06 April 2009 (06.04.2009)

Document type:	Certified copy of priority document	
Document details:	Country/Office: Number: Filing date:	US 61/123,438 07 April 2008 (07.04.2008)
	$\mathcal{O}$	

Date of receipt at the International Bureau: 14 April 2009 (14.04.2009)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse



# AND STREAM AND SYNAMES (D SAN AND STREAM)

### TO ALL TO WILCON THESE PRESENTS SHALL COME:

#### UNITED STATES DEPARTMENT OF COMMERCE

**United States Patent and Trademark Office** 

April 13, 2009

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: 61/123,438 FILING DATE: April 07, 2008 RELATED PCT APPLICATION NUMBER: PCT/US09/39695

THE COUNTRY CODE AND NUMBER OF YOUR PRIORITY APPLICATION, TO BE USED FOR FILING ABROAD UNDER THE PARIS CONVENTION, IS US61/123,438



157239

Certified by

Under Secretary of Commerce for Intellectual Property and Director of the United States Patent and Trademark Office

PR2021-01146 Page 00511

## Systems and Methods for Presenting Information on Mobile Devices

#### Background - The Evolving Requirements for Transforming the Desktop Internet Experience to Mobile Devices

Much has been learned over the last few years as consumers, in remarkably large numbers, have adopted and, with varying degrees of frustration, began using internet-connected mobile devices.

From the failures and limited successes of the past, a clear picture has emerged about the market requirements for effectively transforming content and distributing it to mobile devices. Because this environment has proven to place a very high premium on the time-value of information and on the quality of the user experience, the requirements must include the entire development and delivery platform. Requirements may include:

- Comprehensive Application Model
- Full Authoring Capability with API Extensions that enhance
  - Rapid application development
  - · Much lower maintenance and upgrade costs
  - · Much faster time to market
- Intuitive Navigation
- Advanced Feature Support
  - Messaging and Social Networking Support
    - Full PC Instant Messaging Capability transformed to Mobile
    - Generated and Customized Content
  - Personalization and Temporal Adaptation
    - Consumer Personalization Requests
    - Consumer usage patterns
    - Location
    - Time of day
  - Push Capability
  - Tight Integration with Phone Functions (PIM, Camera, etc.)
  - Mashups
  - Location Awareness integration with all relevant services
- Maximum Distribution to Internet Connected Devices
  - · Removal of most device and network resource constraints
  - Solution for device fragmentation
  - Application Model Support for Operator Extensions and Advanced APIs
  - Robust and Extensible Cross-Platform Application Model
- Complete Analytic Reporting
- Best Possible User Experience