ETSI TS 102 330 V1.1.1 (2004-06)

Technical Specification

Access and Terminals (AT);
Portable Service Format (PSF)
for Interactive Home Devices

Reference

DTS/AT-040007

Keywords

Broadband, IP

ETSI

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

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

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

Important notice

Copyright Notification

Contents

Intellectual Property Rights 5

Foreword 5

Introduction 5

1 Scope 6

2 References 6

3 Definitions and abbreviations 6
3.1 Definitions 6
3.2 Abbreviations 7

4 Overview of a portable service format (informative) 7
4.1 Background 7
4.2 Use of DDE-T for multi-platform services 9
4.2.1 Background 9
4.2.2 Standard UI design elements 11
4.2.3 Navigation control 11
4.2.4 Locator scheme independent of delivery mechanism 11
4.2.5 Synchronization mechanisms 12
4.2.6 Secure access to network resources 12
4.2.7 Text, graphics overlays, and transparency 12
4.2.8 Video resizing and stream selection 12
4.2.9 Sound overlays 13
4.2.10 Presentation of focus 13
4.2.11 Simple animation 13
4.2.12 Testing for the existence of resources 14

5 Portable service format 14
5.1 Markup support 14
5.1.1 Background (informative) 14
5.1.2 HTML mark-up 15
5.2 Style support 15
5.2.1 Background (informative) 15
5.2.2 Style 15
5.3 Scripting elements 16
5.3.1 Background (informative) 16
5.3.2 Scripting 16
5.4 DOM support 16
5.4.1 Background (informative) 16
5.4.2 DOM 17
5.5 DDE-T host object support 17
5.5.1 Background (informative) 17
5.5.2 Host object 17
5.6 Media types 17
5.7 Locator support 17
5.7.1 Background (informative) 17
5.7.2 Locators 17
5.8 Informative guidelines for multi-platform development 17

Annex A (informative): Sample code fragments 19

A.1 Text, graphics overlays, and transparency 19

A.2 Stream selection and presentation 20

A.3 Sound overlays 21

A.4 Typical element usage 22

A.5 Focus usage 23
A.6 Navigation 24
A.7 Animation 27
Annex B (informative): Sample views of Tic Tac Toe game on different target devices 28
B.1 Overview 28
B.2 Tic Tac Toe running on a personal computer browser 29
B.3 Tic Tac Toe running on an a native set top box browser 30
B.4 Tic Tac Toe running on a PDA 31
B.5 Tic Tac Toe running on a mobile phone 32
Annex C (informative): Bibliography 33
History 34

Intellectual Property Rights

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

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

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Access and Terminals (AT).

Introduction

With the proliferation of information devices in the home, there is a growing need for content formats that can be used to author for multiple platforms. Based on both previous standards work and commercial experience, the present document describes the extended use of one such format with a range of home devices.

1 Scope

The present document endorses Declarative Data Essence - Transitional [DDE-T], an existing standardized portable format for interactive television, for use on a wider range of interactive home devices such as home PCs and PDAs.

The present document document also contains informative background on the capabilities of DDE-T, including representative examples of the sample code and figures illustrating the use of key elements.

2 References

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

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

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

[1] SMPTE 397M-2003: "Television - Declarative Data Essence - Transitional".

[2] IETF RFC 2616: "Hypertext Transfer Protocol - HTTP/1.1".

[3] IETF RFC 2818: "HTTP Over TLS".

[4] ISO/IEC 10918-1 (1994): "Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines".

NOTE: JPEG.

[5] ISO/IEC 13818-3 (1998): "Information technology - Generic coding of moving pictures and associated audio information - Part 3: Audio".

NOTE: MPEG.

[6] ISO/IEC 13818-2 (2000): "Information technology - Generic coding of moving pictures and associated audio information: Video".

[7] W3C EC-png: "W3C Recommendation: PNG (Portable Network Graphics) Specification Version 1.0".

[8] IETF RFC 2838: "Uniform Resource Identifiers for Television Broadcasts".

3 Definitions and abbreviations

3.1 Definitions

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

events: instantiation of asynchronous notifications from a client platform to the interactive television content

LaTeX: document preparation system

service: sequence of programs under the control of a broadcaster which can be broadcast as part of a schedule

stream: unidirectional continuous flow of content. Example: MPEG2 video

tuning: act of switching between two MPEG transport streams or multiplexes

user agent: process that interprets HTML documents

3.2 Abbreviations

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

AO	Arguments Object
AV	Audio-Visual
CLUT	Color LookUp Table
CSS	Cascading Style Sheets
DDE-T	Declarative Data Essence - Transitional
DOM	Document Object Model
ECMA	European Computer Manufacturers Association
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transport Protocol
IETF	Internet Engineering Task Force
iTV	interactive TeleVision
JPEG	Joint Picture Expert Group
LaTeX	Lamport TeX
MO	Math Object
MPEG	Motion Picture Expert Group
PAL	Phase Alternate Line
PC	Personal Computer
PDA	Personal Digital Assistant
PDF	Portable Document Format (Adobe)
PNG	Portable Network Graphics
PSF	Portable Service Format
OSD	On Screen Display
SC	Scope Chain
SMPTE	Society of Motion Picture and Television Engineers
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VBScript	Visual Basic Script
W3C	World Wide Web Consortium
WWW	World Wide Web

4 Overview of a portable service format (informative)

4.1 Background

There is an increasing number of home devices such as PCs, interactive television receivers, PDAs that are capable of supporting some type of interactive experience via network connectivity. These devices offer consumers with more choices in how and when they access interactive services. Possible services that could span multiple devices include:

- AV content enhanced with interactive services.
- Information magazines.
- Interactive advertising.
- Gaming.
- eCommerce.
- Customer relationship management and customer care.

- Messaging and communications.
- Simple content guides.
- eGovernment.

Many of these devices, including PCs and PDAs now support some form of web-browser. Interactive television devices and browser-enable mobile phones are more challenging, as many have fewer resources than are available on a PC or PDA. Consequently, cross-platform delivery of these services presents a significant challenge for providers because of the diversity of these devices. Rewriting each service for every platform may be straightforward when targeting similar devices e.g. only PCs or only PDAs. Multi-platform delivery becomes more difficult when addressing platforms with highly disparate capabilities. For example, while all PCs have a Web browser, PDAs often support only restricted browser functionality; low-end set top boxes, which are highly resource constrained, may not be able to support a browser at all. Thus, an authoring approach based on the availability of a full featured resident browser will not reach many potential home devices.

A common approach for multi-platform authors is to use some sort of meta-format to describe a service. A conversion process tied to the authoring environment is then applied to the service description to provide the final application for a particular platform. However, without the use of a common service description format, N authoring environments targeting M platforms leads to N x M conversion tools. A better solution is to have the N authoring environments provide services using a common exchange format, with each of the M target platforms realizing those services via a platform specific conversion step. This reduces the number of conversion steps to of order N + M.

The advantages of using a standard portable format for the exchange of information has been used before for various aspects of document processing in software such as LaTeX; it is also the idea behind Adobe's Portable Data Format. Authors create documents in PDF format and PDF readers are available on each format that convert and render the present document on each device.

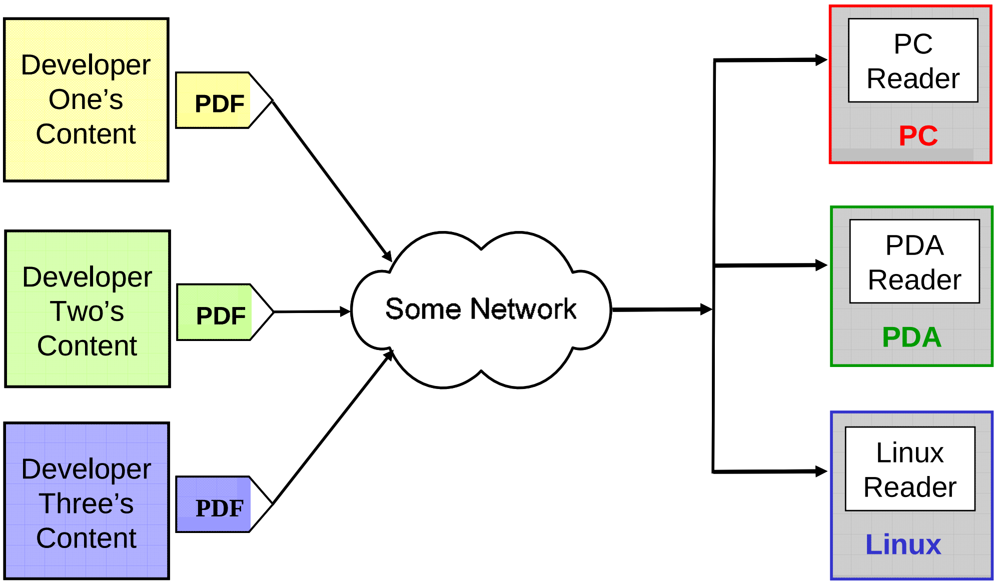


Figure 1: Content distribution using a portable document format

A key feature of the portable document approach is that authors need not worry about the detailed capabilities of the target device(s). Instead, they describe their intended result in a platform independent way. Each target device is responsible for making a best effort to realize that intent within the limitations of the device: images that are too large for a device's screen could be scaled or scrolled; color images may be rendered in black and white, or using a reduced color palette.

One disadvantage of the "reader" model is that it may not be implementable on all intended target devices - smaller footprint devices may be unable to support the minimum set of desired features. Rather than shrink the available feature set, a flexible approach to conversion is necessary to increase the reach of a Portable Service Format (PSF).

Larger footprint devices would continue to use a reader model, implementing a PSF client to render the format on the target device. For less capable devices it is possible that an upstream server may perform the conversion of the content from the standard format to the target device format if the device is not capable of performing this conversion or a range of in-between scenarios. The key to flexible conversion is to have a format that specifies the intended user experience without over-specifying how that experience will be realized.

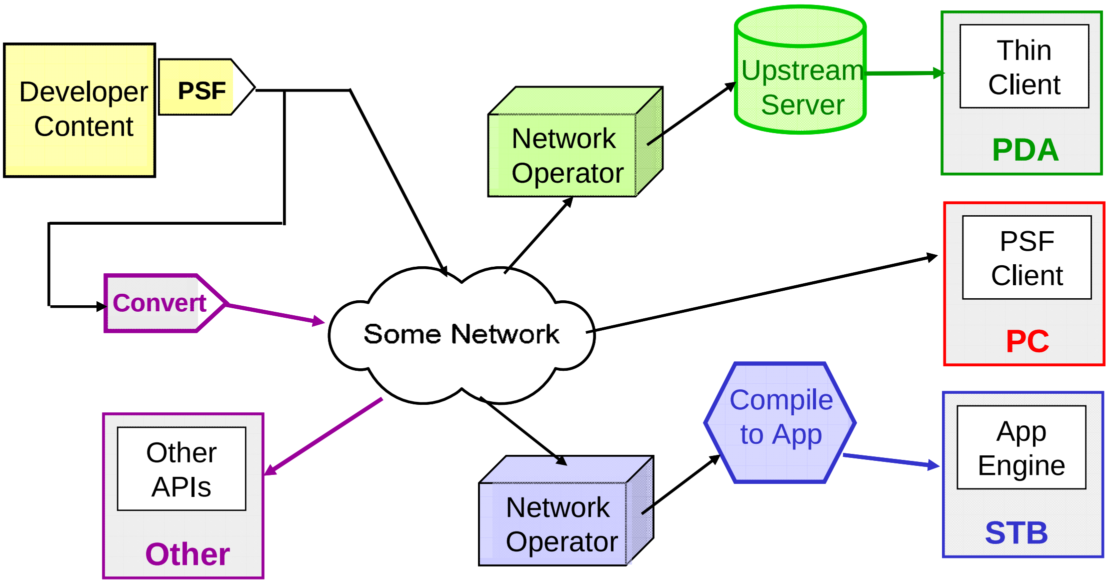


Figure 2: Flexible conversion example

Figure 2 illustrates four different scenarios (although others are possible):

- Source conversion: the content developer converts from the portable format to a specific target before distribution.
- PDF-style: a client capable of running on the target device directly converts the portable format to the target device format.
- Thin client: a network or service operator targeting devices on a specific network maps the portable format functionality onto a client server model, with the target device running a thin client.
- Network compiler: a network or service operator can use an upstream network server to compile the portable format into a native application for the target device.

Flexible conversion can help to support a broader range of devices, network capabilities, and business models.

4.2 Use of DDE-T for multi-platform services

4.2.1 Background

Whilst there is significant variation in platform capabilities between various devices, there are a number of elements that appear frequently in the user experience across devices: user-interface objects such as text, buttons, menus, and forms as well as images and audio/visual streams. In particular, these elements include:

- Text, graphics overlays, possibly with some transparency control.
- Full screen video with some resizing facility.
- Sound overlays.
- Menus, buttons, focus control, links.
- Lists, forms, tables.

- Component and stream selection.
- Simple animation.
- Synchronization of interactive content.
- Ability to test for the presence of return channel and other resources.
- Secure information exchange.
- Locator Scheme independent of delivery mechanism.

A standard format that has these capabilities would be able to address the needs of a large number of these applications.

HTML content for the World Wide Web represents a successful interactive content model: it provides both a relatively portable and familiar way to describe as well as to present interactive pages to users. Combined with style sheets and scripting it provides a rich programming environment that has been used to develop effective interactive services for the World Wide Web. The availability of a large number of developers who have the skills needed to author applications using this platform and inexpensive authoring tools is an important factor in the rapid development of applications and services for this medium. Web content also frequently combines the ease of a mark-up language with procedural scripting to manage both presentation and behavioural logic in an efficient way.

Using interactive television devices as a target device SMPTE initiated a standardization activity to develop and publish an HTML 4.01/XHTML 1.0 content based portable solution for iTV services. This work resulted in the SMPTE-397M [1] standard. DDE-T is based on existing standards from groups such as W3C, ECMA, SMPTE and IETF. While DDE-T is written with a focus on interactive television it was carefully developed with a view to retaining compatibility with existing web-browsers for ease of authoring iTV applications. DDE-T is expected to be supported by a large number of vendors in the iTV field.

Content authored in DDE-T has been demonstrated to work directly with PCs, and with two representative iTV device configurations:

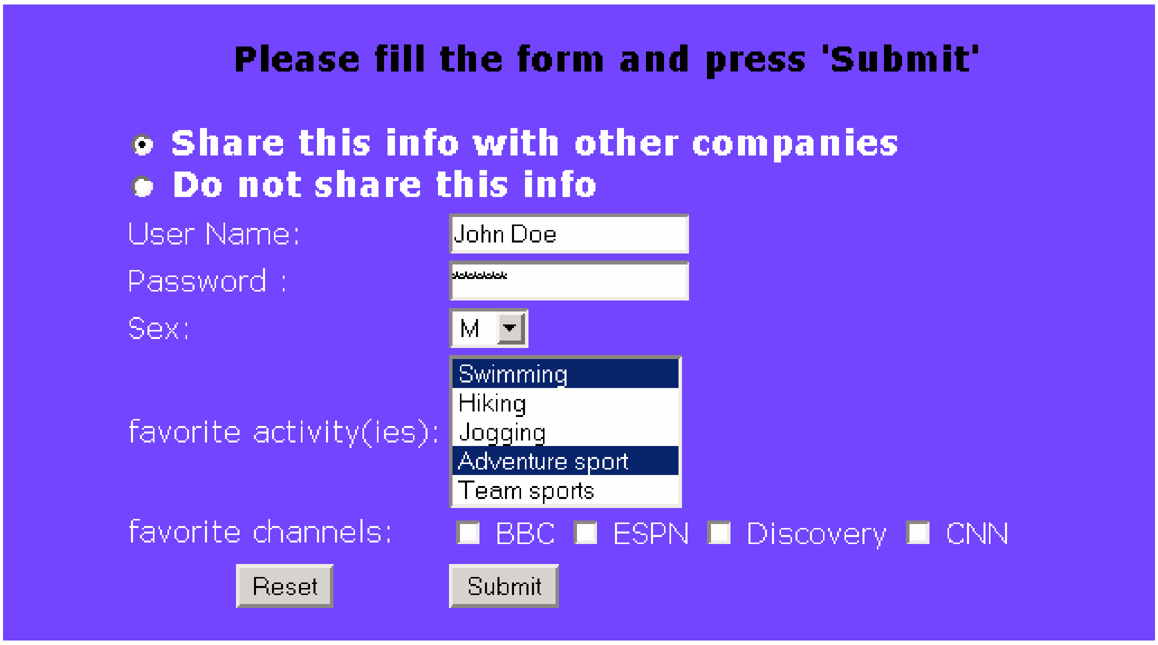
- one based on direct delivery, and
- one using a transcoder.

DDE-T should also be straightforward to deliver to PDAs, either directly (by using a somewhat restricted subset of DDE-T), or via a translator (as with one of the iTV implementations). Based on these results, DDE-T should prove to be a useful standard portable format for home devices. Some examples showing DDE-T running on different platforms are given in annex B.

The present document covers the use of DDE-T in conjunction with a broader range of interactive home devices. It provides a number of informative supplements to the original DDE-T work that should be helpful in making interactive services in the home.

4.2.2 Standard UI design elements

All the standard user interface design elements available in HTML 4.01 can be used in DDE-T, including menus, buttons, lists, links, forms, and tables. See figure 3. The corresponding source code is given in clause A.4.



Please fill the form and press 'Submit'

◉ Share this info with other companies
○ Do not share this info

User Name: John Doe
Password : ******
Sex: M
favorite activity(ies): Swimming, Hiking, Jogging, Adventure sport, Team sports
favorite channels: □ BBC □ ESPN □ Discovery □ CNN

Figure 3: Example using different UI elements

4.2.3 Navigation control

DDE-T has defined a set of style properties to explicitly control the order of navigation through user interface elements on a page. These properties are "dde-nav-index" and "dde-nav-[left,right,top, bottom]". The dde-nav-index value of the target node can be specified as a value of one of the dde-nav-[left,right,top, bottom] in the source node. In addition a compound reference may be used to set the target of the navigation to a node in another frame. Clause A.6 provides a code sample illustrating the use of DDE-T navigation style properties.

Existing user agents that do not support these style properties ignore them as per the CSS requirement that any CSS property that is not understood by the user agent must be ignored. For such user agents, it is possible to achieve the same effect by substituting (or conditionally including) client side script in the page to perform the same function using the procedural syntax in a script. This script could either be created by an author, or generated as part of a transcoding process.

4.2.4 Locator scheme independent of delivery mechanism

DDE-T uses the standard http:// and https:// locators but these locators by themselves do not impose any restriction on the application delivery mechanism. For example in the iTV context the application or parts of the application may be streamed over the broadcast channel. In that situation the broadcast channel acts like a caching proxy as far as the user agent on the home device is concerned. Another example of this is that content may be cached locally on the home device for offline viewing at a later point in time.

4.2.5 Synchronization mechanisms

DDE-T provides authors the ability to control the synchronization of content such as images, video and graphics on a page through the use of the <meta> element to specify the rendering policy.

For example the following fragment would cause the user agent to wait until all the content on the page is loaded (images and graphical elements are loaded, video streams are started) before showing the page to the user:

```
<meta name='dde-render-policy' value='loadComplete'>
```

Other values that DDE-T defines are "layoutComplete" and "progressive" which cause user agents to present the page as soon as the layout is complete or progressively respectively.

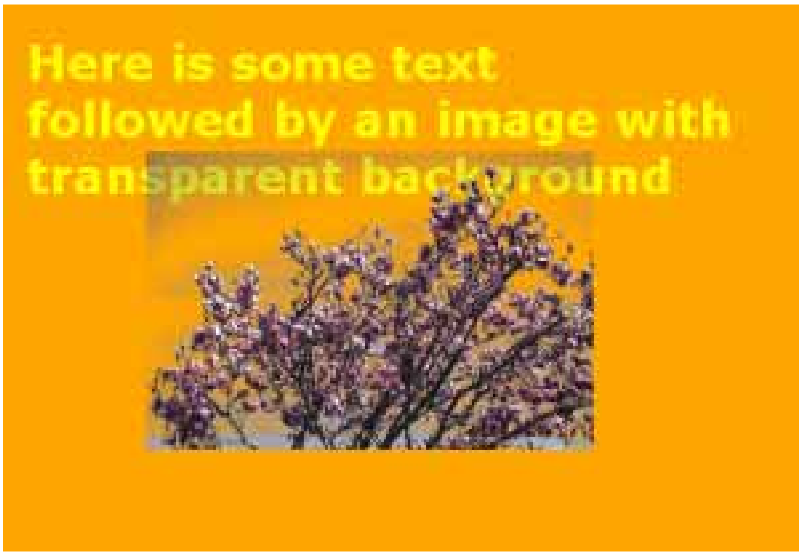
For user agents that do not support this meta tag syntax, a target platform may achieve the same effect via a plug-in exploiting the caching and synchronization available on that platform.

4.2.6 Secure access to network resources

Secure access to network resources in DDE-T is done in the standard way as it is done on the World Wide Web by the use of resources using the https:// locator hosted on secure web servers. Consequently, commonly employed security methods using asymmetric ciphers that use public-key encryption are possible.

4.2.7 Text, graphics overlays, and transparency

HTML 4.01 markup tags can be used in DDE-T for text and graphics overlays. Transparency is currently supported through the use of image formats that support transparency such as GIF or PNG. Future versions of DDE-T may include support for the CSS3 "alpha" property which is part of the CSS-3 specification under development as part of the W3C CSS activity. See figure 4. The corresponding source code is given in clause A.1.



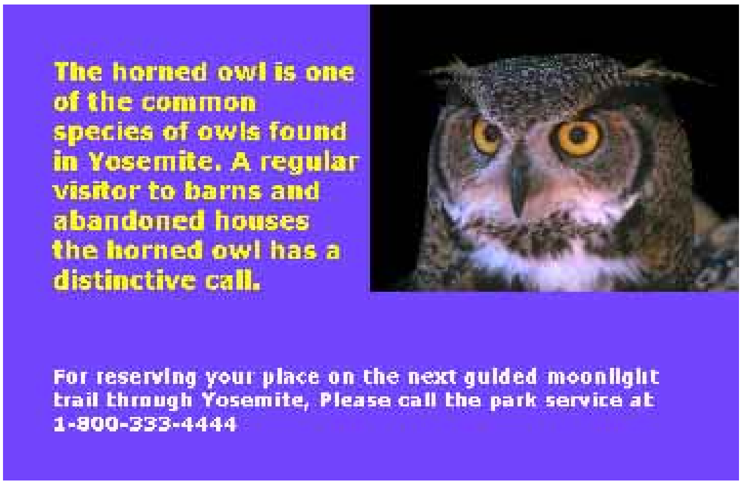
Here is some text followed by an image with transparent background

Figure 4: Example of text and graphics overlay

4.2.8 Video resizing and stream selection

Video can be included in a page using the tv: locator with DDE-T. The size of the <img> or <object> tag that is used to show the video can be used to resize the video. In addition the video may be positioned using CSS absolute positioning. Figure 5 shows an example. The tv: locator syntax also supports selection of specific streams for audio and video, allowing a viewer to, for example, select from the available languages for the audio component. A code example demonstrating this is given in clause A.2.

For existing user agents that do not directly support the tv: locator, there are a number of other possibilities. A given platform could implement equivalent functionality via a plug-in capable of rendering the video resource. Alternatively, the original video material could be transcoded into a format supported by the target platform before delivery. The transcoder would also map the resource pointed to by the tv:// locator to an equivalent streaming video file located on a web server located using the http:// locator.



The horned owl is one of the common species of owls found in Yosemite. A regular visitor to barns and abandoned houses the horned owl has a distinctive call.

For reserving your place on the next guided moonlight trail through Yosemite, Please call the park service at 1-800-333-4444

Figure 5: Example of video presentation

4.2.9 Sound overlays

There is currently no standard cross-platform HTML 4.01 way for playing sounds. Nevertheless, the <embed> tag can be used with suitable plug-ins for playing sound; this mechanism is directly supported by most major user agents. The <embed> tag is deprecated but still available in DDE-T and may be used as shown in the example in clause A.3. That example also illustrates how additional hinting can help with pre-caching of audio resources to deal with possible delays between the delivery of the text and audio.

4.2.10 Presentation of focus

DDE-T syntax uses the standard CSS way to control the visual presentation of focus by use of the focus: pseudo class as specified in the CSS profile for Television. Figure 6 illustrates how the indication of focus can change from element to element. The corresponding code fragment is given in clause A.5.

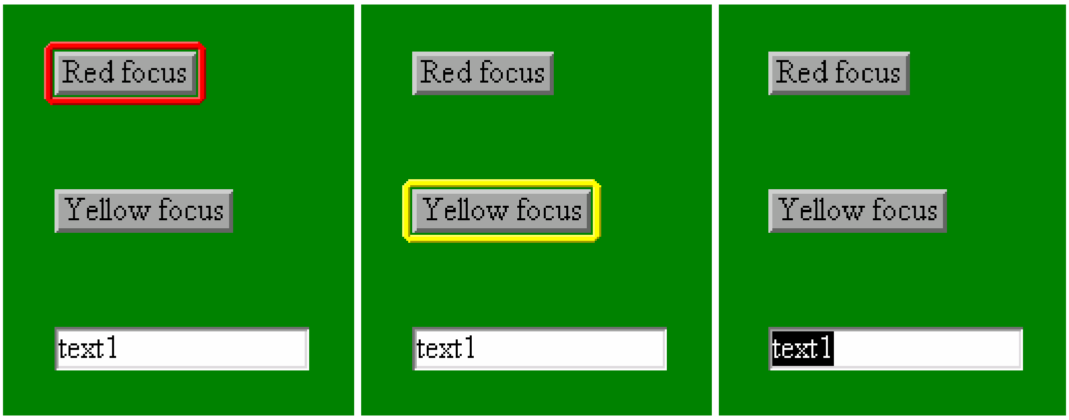


Figure 6: Example of presentation focus

4.2.11 Simple animation

DDE-T relies on scripting support using the ECMA scripts included in a page to achieve simple animations in a page.

4.2.12 Testing for the existence of resources

Testing for the presence of resources such as a return channel is a very common requirement for interactive applications. Nevertheless, due to the diverse nature of possible platform resource, DDE-T does not provide specific testing mechanisms. Rather, authors can use the <object> tag in conjunction with a platform specific component to provide such information. For example, a Java applet or an ActiveX control could be used, depending on the target platform. The object"s properties and methods may themselves be accessed through the script in the page.

5 Portable service format

Unless otherwise noted, the present document includes all elements of the DDE-T [1] specification.

5.1 Markup support

5.1.1 Background (informative)

DDE-T mark-up consists of either XHTML1.0 or HTML 4.01 content containing any of the elements and attributes of the HTML 4.01 specification. As indicated in the present document some HTML 3.2 elements are deprecated but are nevertheless still supported. In summary HTML 4.01 content is largely supported "as is" for DDE-T.

Cross platform application developers can use all the elements and attributes in HTML 4.01 for authoring applications. There are very few additional changes in DDE-T. Application authors have the option to:

1) either ensure that the target platform supports these directly or through a suitable plug-in or an appropriate upstream conversion;

2) can safely ignore these variations;

3) or alternatively authors could include conditional mark-up in their applications for a target platform, but it is recommended that they do so only when no other alternative is available.

These variations are very few and minor as can be seen from the following descriptions.

DDE-T extends the HTML 4.01 syntax with the following additional semantics for iTV applications:

- Using the "tv:" locator to place a video in a page and for stream selection:

```
<img
style='position:absolute;left:10;top:10;width:100;height:100'
src='tv:?audio=AUD0;video=VID0'>
```

  This tag places the live video stream at the specified location and selects the audio stream AUD0 and video stream VID0. Another example is:

```
<a href='tv:?audio=AUD0;video=VID0'> Select Streams </a>
```

  This tag selects the audio and video streams as in the previous example.

- Additional values in the key space for the accessKey html attribute:

```
<input type='button'  value='red key' accessKey='RC_VK_RED'>
```

  The accessKey attribute sets the value to the "RC_VK_RED" which corresponds to the red key on some remote controls for iTV. As mentioned earlier these DDE-T specific features may be mapped to the ones specific to a platform by a transformation tool. In the above example the "red" key may be mapped to a corresponding key (including a soft key) for a hand-held device.

DDE-T has standardized new values for existing mark-up for iTV applications. Currently most web browsers either ignore these tags completely or skip over them if they do not understand them. Hence an application author can use these tags without breaking compatibility with most browsers except that the tag would take effect only on a platform that supports DDE-T. As mentioned before it is possible to map these to corresponding target platform capabilities at the option of the application developer.

- Use of the <META> tag for controlling rendering policy:

  ```
  < META name='dde-render-policy' content='progressive'>
  ```

  This tag indicates that the page will be rendered progressively as content is loaded. Alternative values are "layoutComplete" and "loadComplete" which defer the presentation of the page until the completion of the layout of the page or loading of all content respectively. Similarly DDE-T also introduces the value "dde-render-timeout" for the name attribute and the name-value pair "requested" and "graphics-only" as a hint regarding the content of the page.

- Use of the <LINK> tag for controlling caching:

  ```
  <LINK href='abc.jpg' rel='prerequisite'>
  ```

  This tag indicates that the resource is a pre-requisite for the page. A value of "prefetch" for "rel" means that the resource is requested when the page is loaded but is not a pre-requisite resource.

5.1.2 HTML mark-up

For the avoidance of doubt, clause 5 of DDE-T [1] has been included in the present document.

5.2 Style support

5.2.1 Background (informative)

DDE-T style specification is closely based on the W3C CSS2 specification and the W3C CSS2 profile for TV which is a large and adequate subset of the W3C CSS2 specification. As a result any content written using this subset of features will work on any CSS2 compliant web browser. DDE-T style introduces some new style properties that are useful for iTV applications. These are ignored by standard browsers as per CSS rules and introduce no incompatibility when developing cross platform applications for home devices.

- dde-crop.
- dde-compose-rule.
- dde-nav-index.
- dde-nav-[right, left, top, bottom].
- dde-clut.
- dde-virtual-keyboard.

A cross platform implementation may choose to restrict itself to the common subset of CSS. However it is possible to map these style properties to either platform specific style properties, conditional markup and/or scripting or a combination of the three to achieve the same effect.

5.2.2 Style

For the avoidance of doubt, clause 6 of DDE-T [1] has been included in the present document.

5.3 Scripting elements

5.3.1 Background (informative)

The DDE-T specification supports ECMA 262 (version 3) [1] subject to the restrictions described in ECMA 327 version 3 (see bibliography).

However low-end set top boxes could support a more restricted feature set because of the hardware has performance limitations. These restrictions for the most part do not affect most application developers and there are alternative ways to script in order to achieve the same result.

Thus DDE-T script support provides a useful feature set for cross-platform application development for home devices. For example:

- In order to avoid the necessity of searching the Scope Chain (SC) to resolve a property, support may only be provided for property identification when the object to which the property belongs is explicitly specified.
- The Arguments Object (AO) may be read only.
- The throw and try statements are not available.
- The parseInt(), parseFloat(), and URI handling function properties of the global object may not be available.
- Regular expressions may not be supported. In particular, the RegExp() constructor property of the global object may not be available. The replace() property for Strings need not support regular expressions. (See below for additional restrictions on the replace() property.)
- Errors may not be supported. In particular, the Error(), EvalError(), RangeError(), ReferenceError(), SyntaxError(), TypeError(), and URIError() constructor properties of the global object may not be available.
- The apply() and toString() properties of function objects may not be available.
- The sort() and splice() properties of array objects may not be available.
- The match(), search(), toLocaleLowerCase(), and toLocaleUpperCase() of string objects may not be available. Also, in addition to possibly not accepting a regular expression as a parameter (i.e. for the searchValue or replaceValue parameters) in the replace() property for string objects, an implementation may not support the $replacements. When $replacements are not supported, the replace() method results in a string value derived from the original input string by replacing each matched substring with *newstring*, where *newstring* is the result of converting the replaceValue argument to a string.
- The toExponential() and toPrecision() properties of Number objects need not be available.
- None of the value properties of the Math Object (MO) need be available. Additionally, only the ceil(), floor(), random(), and round() method properties of the Math Object need be available.

5.3.2 Scripting

For the avoidance of doubt, clause 7 of DDE-T [1] has been included in the present document.

5.4 DOM support

5.4.1 Background (informative)

DDE-T DOM Core and HTML support is based on W3C DOM Level 2 specification and is a subset of that specification. This subset was chosen in order to provide a powerful and adequate programming environment for iTV applications while balancing the need to support low-end set-top boxes. As we will see this feature set is also largely supported by most standard browsers on home platforms such as the WebBrowser 2.0 on Palm and NetFront 3.1 on the PocketPC. Hence applications written with DDE-T DOM can run on a large number of home devices.

DDE-T DOM event model is based on the W3C DOM Level 2 Event model and supports dynamic event listener registration and the powerful DOM-2 event dispatch model. DDE-T borrows the textEvent event type from the W3C DOM Level 3 specification for cross-browser and cross-platform support.

5.4.2 DOM

For the avoidance of doubt, clause 7 of DDE-T [1] has been included in the present document.

5.5 DDE-T host object support

5.5.1 Background (informative)

The DDE-T host object model is based on DOM Level 0 which is supported by a majority of existing browsers. This makes DDE-T applications developed with this feature compatible with a majority of web browsers, easing authoring and/or transcoding requirements.

For applications destined for small footprint devices, DDE-T provides additional properties and methods on the navigator host object for cache control. If desired, authors can use conditional scripting to only invoke certain methods for specific platforms.

5.5.2 Host object

For the avoidance of doubt, clause 8 of DDE-T [1] has been included in the present document.

5.6 Media types

The present document supports the use of: MPEG-2 video [6], jpeg [4], png [7], and MPEG-2 audio [5].

5.7 Locator support

5.7.1 Background (informative)

The DDE-T locator scheme supports http: // locators using URIs [8] formulated according to the HTTP 1.1 specification [2]. DDE-T also supports the https:// locator [3], which is useful for designing content for platforms where an interaction channel is available.

DDE-T supports the tv: locators formulated according to the tv: specification [7], including the query syntax for audio and video components.

5.7.2 Locators

For the avoidance of doubt, clause 10 of DDE-T [1] has been included in the present document.

5.8 Informative guidelines for multi-platform development

DDE-T has been specifically crafted to support a wide range of target devices. It is transcodable to enable support for "thin" home devices that are incapable of supporting a full user agent. It supports both user agent identification and scripting to allow for platform-specific adaptations.

Consequently, while DDE-T provides a useful format for targeting multiple platforms, authors should still keep in mind possible major differences between disparate platforms e.g. screen size and resolution.

- Lay out content in such a way as to avoid scrolling on a typical target display. Scrolling can degrade usability of the application. On smaller format devices which lack a keyboard, authors should keep in mind that the full screen may not be available for an input form as part of the display will be occupied by the soft keyboard.

- While different target devices or transcoders may be capable of scaling assets such as images or video, better results may be obtained by bundling multiple versions of critical assets for use on different platforms. For example, screen resolution can vary significantly between different home devices. While a set top box frequently supports 720 × 576 pixels (PAL), PDA resolutions may be only 229 × 255 pixels or 320 × 320 pixels.

- When targeting multiple platforms, server side scripting, when used in conjunction with browser detection and dynamic page generation, can assist in matching content to the target platform.

- Consider the use of client side conditional code if server side scripting cannot address differences in platforms:

    - Client side scripts should be used more sparingly because they add to the size and the download times for a page.

Annex A (informative): Sample code fragments

A.1 Text, graphics overlays, and transparency

The following code fragment illustrates the use of DDE-T syntax to produce text and graphics overlays along with transparency. The code below produces a text and a PNG image overlay on top of a solid orange background. The PNG image has some parts that are transparent and the background orange color shows through in these areas.

```
<html>
<style>
.bodyclass
{
    font-family:Verdana;
    font-size:24pt;
    font-style:normal;
    font-weight:bold;
    color:#FFFF00;
    background-color:orange;
}
</style>
<body class="bodyclass" style=" width:720; height:576">
<div style="position:absolute; left:100; top:100; width:500; height:400">
    <div style="position:absolute; left:-80; top:-80">
    Here is some text <br>
    followed by an image with transparent background<br>
    </div>
    <img src="resource/MagnoliaAlpha.png" style="position:absolute; left:0; top:0; width:300;
height:200;">
<div>
</body>
</html><html>
    <body style='width:720; height:576;'>
                   <div style='
</html>
```

A.2 Stream selection and presentation

The following code example shows how to use DDE-T syntax to scale and position the video in the top right corner and present it with some text. The tv: locator is used to locate the video and in this example the video presented (VID1) could be one of the camera angles in the program and the audio presented (FR) could be the French language audio. This example also shows how scripting may be used to present a still image in place of the AV stream if the resource is not available.

```
<html>
<style>
.bodyclass
{
    font-family:Verdana;
    font-size:18pt;
    font-style:normal;
    font-weight:bold;
    color:white;
    background-color:#7744FF;
}

</style>

<script>
function on_load_handler()
{
    if ( navigator.appName != "H2O" )
    {
        document.getElementById("VIDEO").style.backgroundImage = "url(resource/OwlAlpha.png)" ;
    }
    else
    {
        document.getElementById("VIDEO").style.backgroundImage =
"url(tv://national_geographic.com/yosemite/owl?audio=FR;video=VID1)" ;
    }

}
</script>

<body class="bodyclass" style=" width:720; height:576;" onload="on_load_handler();">
    <!-- for the quarter screen video -->
    <div style="position:absolute; background-color:black; left:360; top:0; width:360; height:280">
        <img class="video" id="VIDEO" style="position:absolute; left:0; top:0; width:360;
height:280" src="resource/OwlAlpha.png">
    </div>

    <div style="position:absolute; color:yellow; left:50; top:50; width:300; height:500">
    The horned owl is one of the common species of owls found in Yosemite.
    A regular visitor to barns and abandoned houses the horned owl has a distinctive call.
    </div>

    <div style="position:absolute; color:white; font-size:14pt; left:50; top:350; width:600;
height:200">
    For reserving your place on the next guided moonlight trail through Yosemite,
    Please call the park service at 1-800-333-4444
    </div>

</body>
</html>
```

A.3 Sound overlays

This example demonstrates the use of sound overlays via the <embed> tag. This example also makes use of two other DDE-T syntax elements. The first is the use of the <link> element as a cache hint to the user agent. The second one is the content hint using the <meta> tag.

A user agent that honors the DDE-T syntax for <link> element will intelligently pre-cache the sound clip files providing better performance when the content is delivered over channels with long latencies such as in a broadcast environment. The <meta> tag is a hint to the user agent that indicates that live video is not included in this page. The user agent may use this hint when it is running on a low-end iTV receiver and allocate resources for OSD display accordingly. Existing user agents harmlessly ignore these two syntax, but transcoding to equivalent platform specific directives that may be passed to the user agent as <meta> tags is possible.

```
<html>
<link href="bark.mp2" rel="prerequisite">
<link href="bummer.mp2" rel="prerequisite">
<meta name="requested" content="graphics-only" >
<script language="javascript">

function play_bark()
{
    document.getElementById("bark").play();
}

function play_klang()
{
    document.getElementById("klang").play();
}

</script>

<style>
:focus{
outline-color:gold;
outline-width:4px;
}
</style>

<body
style="dde-clut:url(resource/clut_poker.clt);color:blue; background-
color:lightblue;width:720;height:576; "
id="oID_1" width="720" height="480">

<!-- This is an example of sound overlays -->
<embed id="bark" src="resource/bark.mp2" hidden="true" autostart="false">
<embed id="klang" src="resource/klang.mp2" hidden="true" autostart="false">

<!-- buttons to play the sound -->
<input type="button" value="bark" onclick="play_bark();">
<br>
<input type="button" value="klang" onclick="play_klang();">
<br>
<br>

<!-- anchors to play the sound -->
<a href="resource/bark.mp2"> Click here to hear the 'bark' sound </a>
<br>
<br>
<a href="resource/klang.mp2"> Click here to hear the 'klang' sound </a>
<br>
<br>

</body>
</html>
```

A.4 Typical element usage

The following code fragment illustrates the use of typical HTML 4.0.1 design elements on a DDE-T page. The on-screen presentation of this code is shown in figure 3.

```
<html>
<style>
.bodyclass
{
    font-family:Verdana;
    font-size:14pt;
    font-style:normal;
    font-weight:bold;
    color:white;
    background-color:#7744FF;
}
</style>
<body class="bodyclass"
style="dde-clut:url(resource/clut_poker.clt);width:720;height:576;"
id="oID_1" width="720" height="480">
<br>
<br>
<center style="color:black;"> Please fill the form and press 'Submit' </center>
<!-- This is an example UI design elements -->
    <form action="submit.asp" method="get">
        <div style="position:absolute; left:100; top:100;">
            <table>
              <tr>
                <input type="radio" name="agree" value="1" style="font-size:12pt;" >
                            Share this info with other companies<br>
                <input type="radio" name="agree" style="font-size:12pt;">
                            Do not share this info
              </tr>
              <tr>
                <td>
                    User Name:
                </td>
                <td>
                    <input type="text" value="John Doe">
                </td>
              </tr>
              <tr>
                <td>
                    Password :
                </td>
                <td>
                    <input type="password" value="secret">
                </td>
              </tr>
              <tr>
                <td>
                    Sex:
                </td>
                <td>
                    <select>
                    <option> M </option>
                    <option> F </option>
                    </select>
                </td>
              </tr>
              <tr>
                <td>
                    favorite activity(ies):
                </td>
                <td>
                    <select multiple>
                    <option> Swimming </option>
                    <option> Hiking </option>
                    <option> Jogging </option>
                    <option> Adventure sport </option>
                    <option> Team sports </option>
                    </select>
                </td>
              </tr>
              </tr>
```

```
                <td>
                    favorite channels:
                </td>
                <td>
                    <input type="checkbox"> BBC
                    <input type="checkbox"> ESPN
                    <input type="checkbox"> Discovery
                    <input type="checkbox"> CNN
                </td>
              </tr>
              <tr>
              </tr>
              <tr>
              </tr>
              <tr>
                <td>
                    <center>
                    <input type="reset" value="Reset">
                    </center>
                </td>
                <td>
                    <input type="submit" value="Submit">
                </td>
              </tr>
            </table>
        </div>
    </form>
</body>
</html>
```

A.5 Focus usage

The following code fragment demonstrates the DDE-T syntax to control focus. The first button would indicate the focus using a red outline the second button would use a yellow outline and the text below has no outline to indicate focus, although it may use other types of visual feedback to indicate focus such as selecting the text when the user navigates to the text field.

```
<html>
<style>
:focus
{
  outline-width:0px;
}

:focus#button_red_focus
{
  outline-width:4px;
  outline-color: red;
}

:focus#button_yellow_focus
{
  outline-width:4px;
  outline-color: yellow;
}
</style>
<body style="color:white; background-color:green; font-family:Verdana; font-size:14pt;">
    <div style="position:absolute; left:100; top:100;">
        <input id="button_red_focus" type="button" value="Red focus"><br><br><br>
        <input id="button_yellow_focus" type="button" value="Yellow focus"><br><br><br>
        <input type="text" value="text1"><br>
    </div>
</body>
</html>
```

A.6 Navigation

DDE-T has defined a set of style properties to explicitly control the order of navigation through user interface elements on a page. The following code fragments illustrate the use of these style properties to control navigation in a page. The file nav_frame.htm contains two frames nav1.htm and nav2.htm which define the navigation rules using the dde-nav-index and dde-nav-[left,right,top, bottom] style properties:

File nav_frame.htm

```
<!—File1: nav_frame.htm -->
<html>
<head>
<title> Nav Properties (accesskey, dde-nav-index, dde-nav- left, right, up, down) </title>
</head>

<frameset cols= "30%, *">
    <frame name="nav1" id="nav1" frameborder="no" marginwidth="0" marginheight="0" noresize
scrolling="no" src="nav1.htm">
    <frame name="nav2" id="nav2" frameborder="no" marginwidth="0" marginheight="0" noresize
scrolling="no" src="nav2.htm">
</frameset>
</html>
```

File nav1.htm

```
<!—File1: nav1.htm -->

<html>

<style>
:focus{
outline-width:7px;
outline-color:magenta;
}
</style>

<script language="javascript">

</script>

<body style="dde-clut:url(resource/logo.clt); background-color:lightblue; color:black;
width:720;height:576;"
>
<div style="position:absolute;left:25;top:10;right:25">

<br>
<br>
    <INPUT id="btn1" type="button" value="     One "
    style="width:100;color:black;background-color:white;
    dde-nav-index:1;
    dde-nav-right:'nav2#btn2';
    dde-nav-left:'nav2#$1';
    dde-nav-up:5;
    dde-nav-down:'#btn5';
    "
    accesskey="1"
    >
<br>
<br>
    <INPUT id="btn2"type="button" value="      Two"
    style="width:100;color:black;background-color:white;
    dde-nav-index:2;
    dde-nav-right:'nav2#btn1';
    dde-nav-left:'nav2#$2';
    "
    accesskey="2"
    >
<br>
<br>

    <INPUT id="btn3" type="button" value="    Three"
    style="width:100;color:black;background-color:white;
    dde-nav-index:3;
    dde-nav-right:'nav2#btn4';
    dde-nav-left:'nav2#$3';
    "
    accesskey="3"
    >
<br>
<br>
    <INPUT id="btn4" type="button" value="   Four"
    style="width:100;color:black;background-color:white;
    dde-nav-index:4;
    dde-nav-right:'nav2#btn3';
    dde-nav-left:'nav2#$4';
    "
    accesskey="4"
    >
<br>
<br>
    <INPUT id="btn5" type="button" value="  Five"
    style='width:100;color:black;background-color:white;
    dde-nav-index:5;
    dde-nav-right:"nav2#btn5";
    dde-nav-left:"nav2#$5";
    dde-nav-up:"#btn1";
    dde-nav-down:5;
    '
    accesskey="5"
    >
<br>
<br>

</div>
</body>
</html>
```

Main body of code

```
<html>

<script language="javascript">

</script>

<body style="dde-clut:url(resource/logo.clt); background-color:blue; color:black;
width:720;height:576;
key-list:numeric_set,scroll_set,navigation_set;
"
>
<div style="position:absolute;left:25;top:10;right:25">

<br>
<br>
    <INPUT id="btn1" type="button" value="     One "
    style="width:100;color:black;background-color:white;
    dde-nav-index:1;
    dde-nav-right:'nav1#btn1';
    dde-nav-left:'nav1#$1';
    dde-nav-up:'#$5';
    dde-nav-down:'#btn5';
    "
    accesskey="1"
    >
<br>
<br>
    <INPUT id="btn2"type="button" value="      Two"
    style="width:100;color:black;background-color:white;
    dde-nav-index:2;
    dde-nav-right:'nav1#btn3';
    dde-nav-left:'nav1#$2';
    "
    accesskey="2"
    >

<br>
<br>
    <INPUT id="btn3" type="button" value="    Three"
    style="width:100;color:black;background-color:white;
    dde-nav-index:3;
    dde-nav-right:'nav1#btn2';
    dde-nav-left:'nav1#$3';
    "
    accesskey="3"
    >
<br>
<br>
    <INPUT id="btn4" type="button" value="   Four"
    style="width:100;color:black;background-color:white;
    dde-nav-index:4;
    dde-nav-right:'nav1#btn5';
    dde-nav-left:'nav1#$4';
    "
    accesskey="4"
    >
<br>
<br>
    <INPUT id="btn5" type="button" value="  Five"
    style="width:100;color:black;background-color:white;
    dde-nav-index:5;
    dde-nav-right:'nav1#btn4';
    dde-nav-left:'nav1#$5';
    dde-nav-up:'#btn1';
    dde-nav-down:'#$1';
    "
    accesskey="5"
    >
<br>
<br>

</div>
</body>
</html>
```

A user agent may render this page as shown in figure A1. The arrows indicate the navigation sequence for the first button on the page.

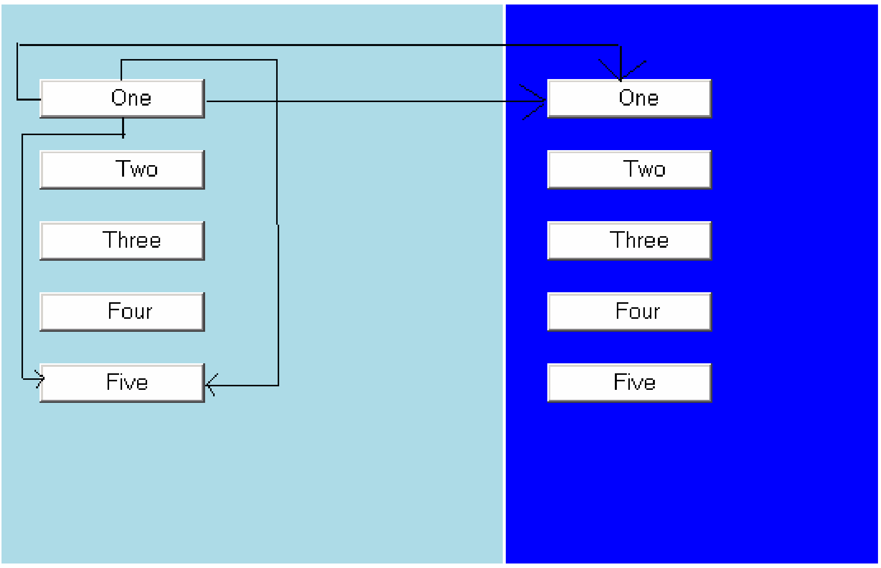


Figure A.1: example of navigation flow

A.7 Animation

The following script fragments shows how a simple animation is implemented using a script.

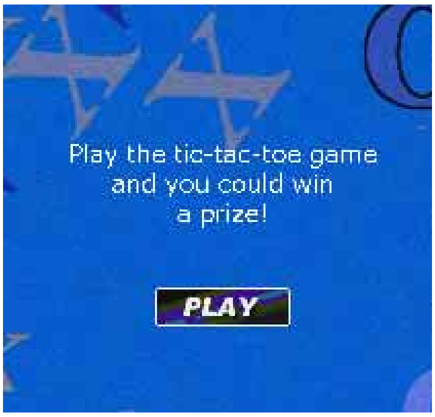
```
var animation_timer_started;
var animation_timer_id;
var animation_curr_picture;

animation_timer_started = false;
animation_timer_id = null;
animation_curr_picture = 1;

function toggle_animation_timer()
{
    if ( animation_timer_started == true )
    {
        window.clearInterval( animation_timer_id );
        animation_timer_started = false;
        animation_curr_picture = 1;
        dog_anim.style.visibility = "hidden";
        anim_bg.style.visibility = "hidden";
    }
    else
    {
        animation_timer_id = window.setInterval( animation_handler, 200 );
        animation_timer_started = true;
    }
}

function animation_handler()
{
    dog_anim.style.visibility = "visible";
    anim_bg.style.visibility = "visible";
    img_str = "resource/dogrun_anim_" + String(animation_curr_picture) + ".gif";
    dog_anim.src = img_str;

    animation_curr_picture++;
    if ( animation_curr_picture == 36 )
        animation_curr_picture = 1;
}
```

Annex B (informative): Sample views of Tic Tac Toe game on different target devices

B.1 Overview

The TicTacToe game is a simple application authored using DDE-T. This application uses HTML 4.01 markup, ECMA scripting and the DOM and CSS features from the DDE-T specification. The game is developed as a series of .asp pages. The application uses server-side scripting (VBScript) with browser detection, as well as conditional client-side scripts to target multiple home device platforms.

The game consists of a "Welcome" screen with a single "Play" button. When the user presses the "Play" button the the "Game" screen is presented to the user.



Play the tic-tac-toe game
and you could win
a prize!

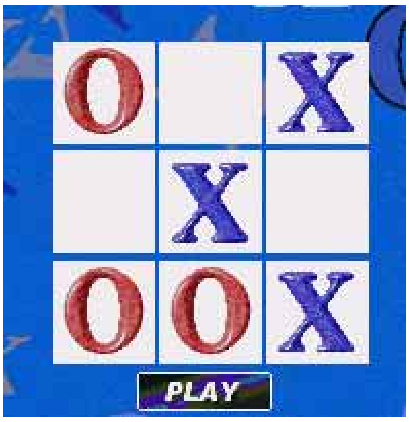
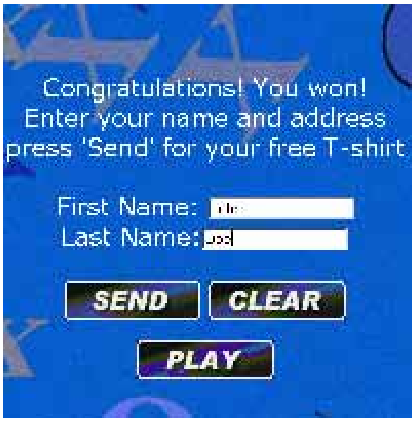


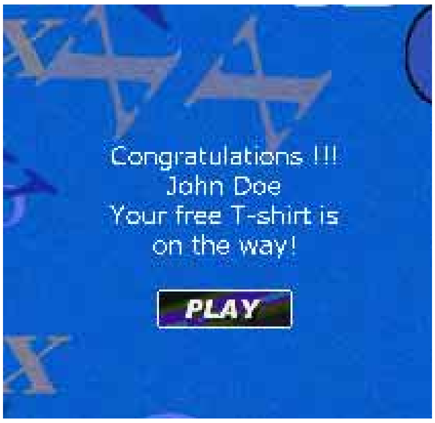
Figure B.1 **Figure B.2**

If the user wins the game he is presented with a form to fill and submit to a backend server. When the user submits the form to the server, it generates a confirmation page dynamically using the values of text fields entered by the user.



Congratulations! You won!
Enter your name and address
press 'Send' for your free T-shirt

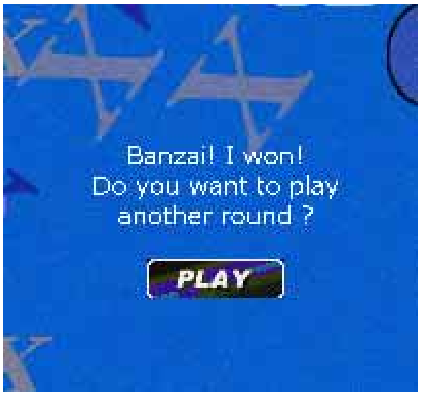
First Name:
Last Name:



Congratulations !!!
John Doe
Your free T-shirt is
on the way!

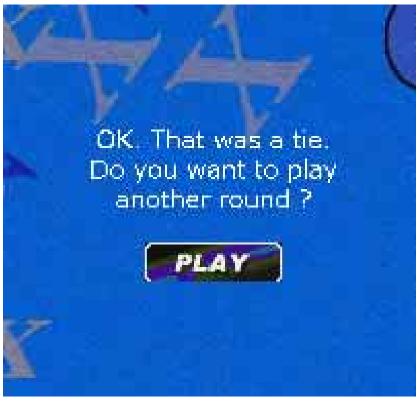
Figure B.3 **Figure B.4**

If the user loses the game or it is a tie he is given the option to play the game again.



Banzai! I won!
Do you want to play
another round ?

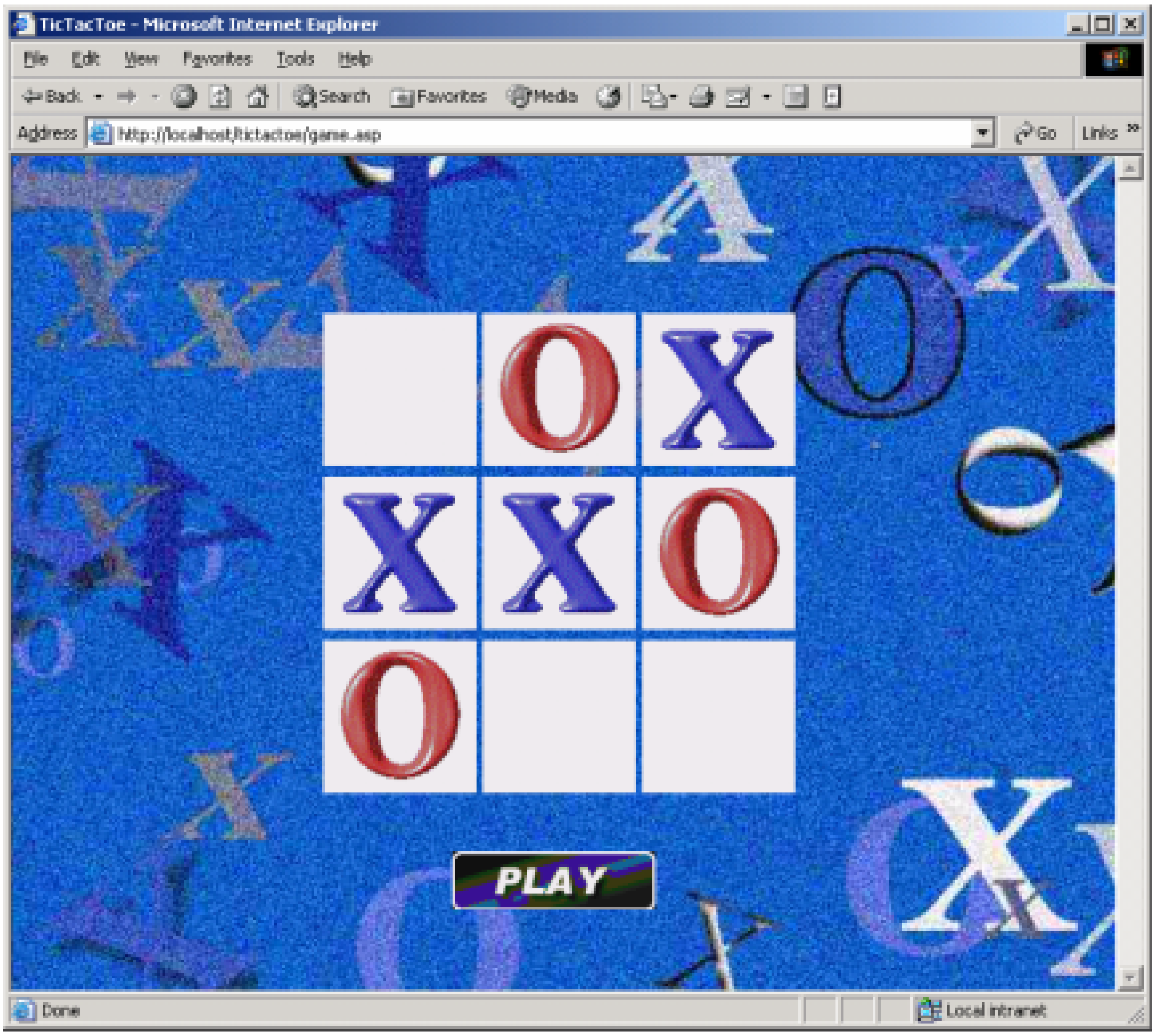
Figure B.5



OK. That was a tie.
Do you want to play
another round ?

Figure B.6

B.2 Tic Tac Toe running on a personal computer browser



TicTacToe - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/tictactoe/game.asp
Go Links
O X
X X O
O
PLAY
Done
Local intranet

Figure B.7

B.3 Tic Tac Toe running on an a native set top box browser

Figure B.8 shows the Tic Tac Toe game running on a set top box implementing a DDE-T compatible browser. Such a browser can directly convert the DDE-T markup and render it on the set-top box.

Figure B.8 also demonstrates a common multi-platform authoring issue: some home device browsers may allocate part of the display to a toolbar and/or a softkeyboard, reducing the available screen area.

Application developers need to take into account all these factors while targetting an application to multiple home device platforms. Server side scripting can be useful to address such differences in screen layout as well as differences due to variations of the form factor.

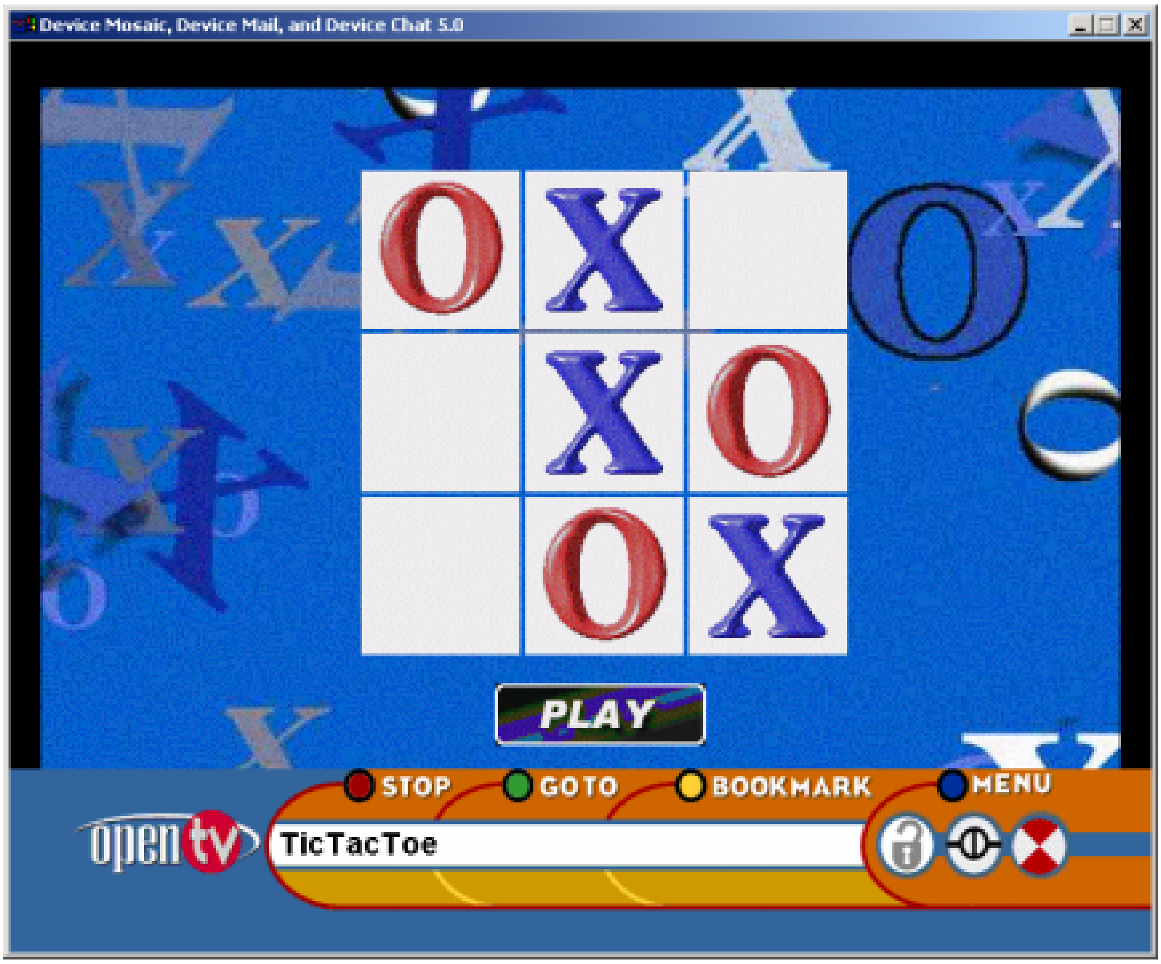


Figure B.8

B.4 Tic Tac Toe running on a PDA

Personal digital assistants are another popular information device. The following figure provide a screen shot of the Tic Tac Toe application running on a simulator for a popular PDA platform which supports modern web browsers. DDE-T applications such as the tictactoe game app can run on these devices directly.

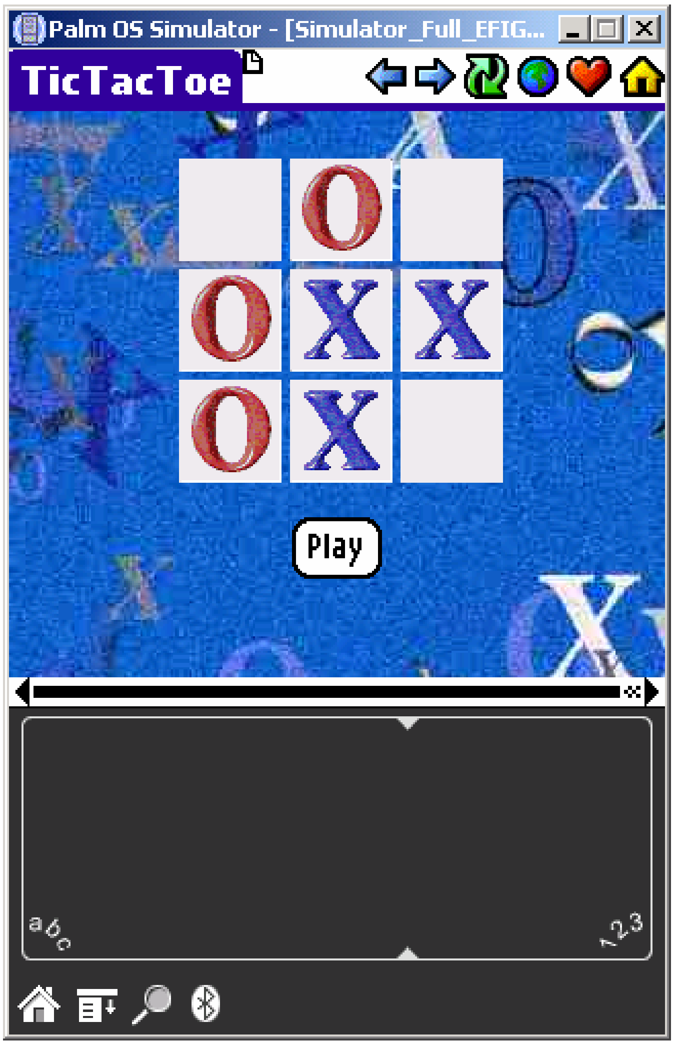


Figure B.9

B.5 Tic Tac Toe running on a mobile phone

Newer cellular phones increasingly support modern browser technologies. Those supporting HTML 4.0.1 can run DDE-T applications directly. The figure below shows the tictactoe game running in a typical mobile phone environment.

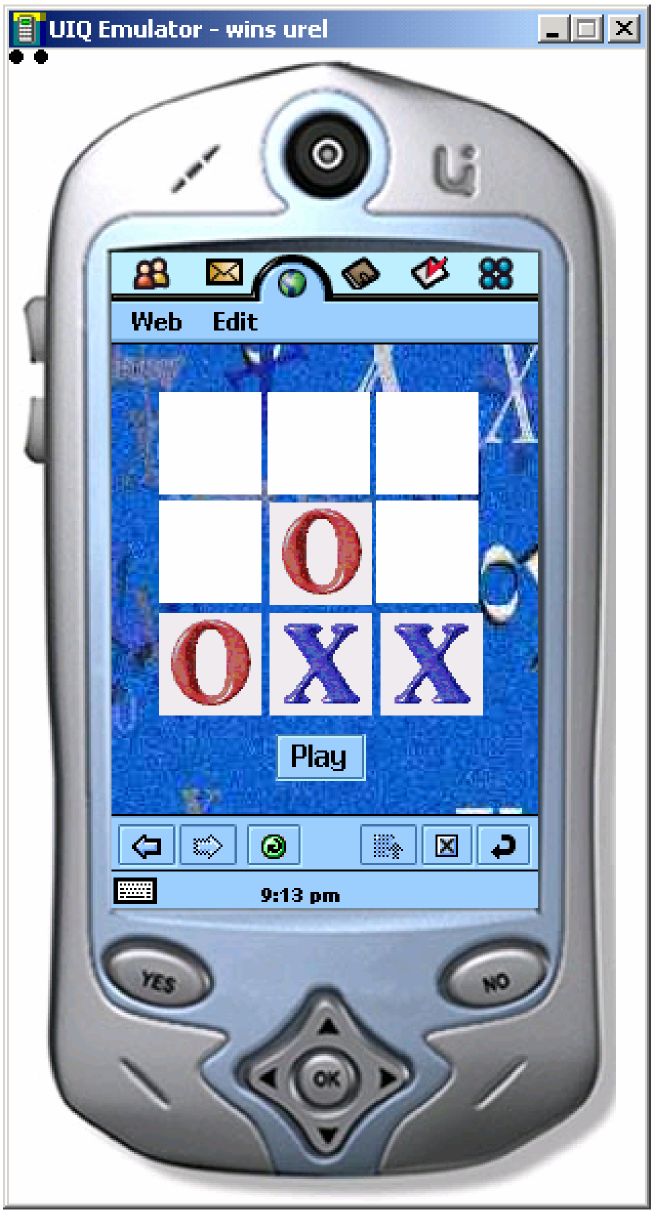


Figure B.10

Annex C (informative): Bibliography

- W3C CR-css-tv-20020807: "W3C Candidate Recommendation: CSS TV Profile 1.0".
- SMPTE 366M-2002: "Television - Declarative Data Essence- Document Object Model Level 0 (DOM-0) and Related Object Environment".
- W3C REC-DOM-Level-2-Core-20001113: "W3C Recommendation: Document Object Model (DOM) Level 2 Core Specification Version 1.0".
- W3C REC-DOM-Level-2-Style-20001113: "W3C Recommendation: Document Object Model (DOM) Level 2 Style Specification Version 1.0".
- W3C REC-DOM-Level-2-HTML-20030109: "W3C Recommendation: Document Object Model (DOM)Level 2 HTML Specification Version 1.0".
- W3C REC-DOM-Level-2-Events-20001113: "W3C Recommendation: Document Object Model (DOM) Level 2 Events Specification Version 1.0".
- ECMA-262: "ECMA Standard: ECMAScript 3rd edition".
- ECMA-327: "ECMA Standard: ECMAScript 3rd edition, Compact Profile".
- W3C REC-html401-19991224: "W3C Recommendation: HTML 4.01 Specification".

History

Document history		
V1.1.1	June 2004	Publication