International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.351

(06/2008)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Directory services architecture for audiovisual and multimedia services

Semantic web interface for multimedia terminal and system directories (SWIM-D)

Recommendation ITU-T H.351

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	**H.350–H.359**
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND AND TRIPLE-PLAY MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	H.620–H.629
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T H.351

Semantic web interface for multimedia terminal and system directories (SWIM-D)

Summary

Recommendation ITU-T H.351 describes a semantic web interface in XHTML that can be used to provide directory lookup and menu capabilities for multimedia terminals and systems.

Source

Recommendation ITU-T H.351 was approved on 13 June 2008 by ITU-T Study Group 16 (2005-2008) under Recommendation ITU-T A.8 procedure.

Keywords

Address book, directory, multimedia, web interface, white pages.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Introduction

This Recommendation describes a method by which XHTML documents can be accessed by multimedia terminals, systems, web browsers and middleware systems to recognize, navigate, process and display white pages directory structures. It addresses three primary needs:

- **Representing both people and resources in the directory**. While people are represented in directories in well known formats, there are not good standards for representing resources in the directory, even though those resources have important contact information. For example "help desk" and "conference room 112" are important entities which may contain multimedia conferencing resources that need to be contacted.
- **Diverse display characteristics of systems**. Some multimedia terminals, such as IP telephones, have limited user interface and processing capabilities, necessitating a simplified markup scheme for displaying address books and directory information. Advanced terminals, such as web browsers or video conferencing terminals, may have rich display capabilities yet need to formulate a directory display from the same data.
- **Structured navigation**. It may be that an organization wishes to control the way that white pages applications are served out to terminals and systems. For example, an organization may wish to publish an address book that allows navigation by department, product line, or individual last name.

Recommendation ITU-T H.351

Semantic web interface for multimedia terminal and system directories (SWIM-D)

1 Scope

This Recommendation describes a markup vocabulary for XHTML documents representing white pages directory information for multimedia terminals and systems. It describes a uniform resource locator (URL) syntax that supports directory searching. It also enumerates client and server behaviours necessary to support SWIM-D interaction effectively. The purpose of this Recommendation is to provide a framework for structured directory browsing when the underlying data structures are not known, such as when publishing directories of people, places and virtual organizations.

This Recommendation does not describe directory structures or directory access protocols.

This Recommendation is related to other ITU-T H.350-series Recommendations. While Recommendations ITU-T H.350.1 through H.350.7 describe directory schemas for various protocols that can drive call signalling engines directly, this Recommendation is intended for higher level access to directory data for white pages browsing only. [b-ITU-T E.115] describes a computerized directory assistance protocol for information exchange among service providers in support of assistance/inquiry services.

White pages implementers should review all of these related Recommendations for applicability.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[IETF RFC 2616] IETF RFC 2616 (1999), *Hypertext Transfer Protocol – HTTP/1.1.*
<http://www.ietf.org/rfc/rfc2616.txt>

[IETF RFC 2617] IETF RFC 2617 (1999), *HTTP Authentication: Basic and Digest Access Authentication*.
<http://www.ietf.org/rfc/rfc2617.txt>

[IETF RFC 2818] IETF RFC 2818 (2000), *HTTP Over TLS*.
<http://www.ietf.org/rfc/rfc2818.txt>

[IETF RFC 3986] IETF RFC 3986 (2005), *Uniform Resource Identifier (URI): Generic Syntax*.
<http://www.ietf.org/rfc/rfc3986.txt>

[W3C XHTML 1.1] W3C Recommendation (2001), *XHTML™ 1.1 – Module-based XHTML.*
<http://www.w3.org/TR/2001/REC-xhtml11-20010531/>

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 cascading style sheets (CSS): [b-W3C HTML 4.01] A simple mechanism for adding style (e.g., fonts, colours, spacing) to web documents (W3C).

3.1.2 extensible markup language (XML): [b-W3C XML 1.0] A generalized text format maintained by the World Wide Web Consortium designed as a meta language to describe other markup languages.

3.1.3 hypertext markup language (HTML): [b-W3C HTML 4.01] The markup language for the world wide web maintained by the World Wide Web Consortium.

3.1.4 lightweight directory access protocol (LDAP): [b-IETF RFC 3377] A directory access protocol defined by the Internet Engineering Task Force to interact with X.500 directories.

3.1.5 XHTML: [W3C XHTML 1.1] An XML-compliant version of HTML designed to be syntactically rigorous in order to support machine-to-machine communication. Maintained by the World Wide Web Consortium.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 client: An agent that can access a directory page. A client can be a standard HTML browser or a SWIM-D client.

3.2.2 directory: A canonical collection of attributes about people and resources in an organization, including both the enterprise directory and the endpoints directory.

3.2.3 directory page: An XHTML document containing directory information. Directory pages may provide navigation, search and results in addition to other information.

3.2.4 directory proxy: A web server that publishes SWIM-D directory pages. Directory proxies have deep knowledge about the directories they access, and format that information into meaningful and useful directory pages for clients.

3.2.5 endpoint directory: A directory of endpoint information when that information is stored in a standalone directory, such as is the case with [b-ITU-T H.35x].

3.2.6 enterprise directory: A directory of people and resources within an organization.

3.2.7 resource: A non-human entity represented in a directory that may have with it associated real-time communications account and address data. Resources may be physical, such as "conference room 12" or virtual, such as "help desk", each of which has a voice or video over IP address.

3.2.8 semantic web interface for multimedia directories (SWIM-D): The interface that defines the structure of directory pages and how to access them via their uniform resource locator (URL).

3.2.9 SWIM-D client: A client that is capable of understanding and interpreting SWIM-D directory pages. A SWIM-D client can be an HTML browser that has knowledge of SWIM-D, or it can be an XML-aware agent that processes directory pages for non-HTML purposes, such as an address book or auto-dialler.

3.2.10 SWIM-D directory page: A directory page that is marked up using SWIM-D metadata.

3.2.11 SWIM-D URL: A uniform resource locator (URL) in the format specified to support access to SWIM-D directory pages.

3.2.12 web browser: A software application capable of rendering HTML and XHTML documents.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CSS	Cascading Style Sheets
DNS	Domain Name System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
LDAP	Lightweight Directory Access Protocol
SQL	Structured Query Language
SWIM-D	Semantic Web Interface for Multimedia Directories
URL	Uniform Resource Locator
XML	Extensible Markup Language
VoIP	Voice over Internet Protocol

5 Conventions

In this Recommendation, the following conventions are used:

- "Shall" indicates a mandatory requirement.
- "Should" indicates a suggested but optional course of action.
- "May" indicates an optional course of action rather than a Recommendation that something take place.

References to clauses, annexes and appendices refer to those items within this Recommendation, unless another specification is explicitly listed.

6 The problem of resource representations in directories

In applications such as voice over IP, video conferencing and instant messaging it is sometimes desirable that many different types of resources have communication accounts. Some examples are listed below:

- **People**: People may have a SIP or H.323 account which they use for voice over IP.
- **Conference rooms**: Conference rooms may exist which contain H.323 video conferencing equipment to allow for collaboration.
- **Classrooms**: Classrooms may have several different types of audio and video conferencing equipment all operating simultaneously, such as SIP audio conferencing and H.323 video conferencing.
- **Surveillance sites**: H.323, SIP or other remote observation cameras may be deployed throughout an organization to which security staff can connect for remote observation.
- **Medical equipment**: A medical cart may contain multimedia conferencing equipment, along with medical diagnostic equipment. This cart may be mobile and be checked out many times per day and used throughout a hospital.
- **Virtual organizations**: It is desirable for a hospital to be able to publish separate numbers for "main hospital number", "emergency", "paediatrics", "paediatric nursing", and "paediatric doctor on call". Each of these virtual organizations may have a VoIP account

which is published in both printed and on-line directories, and for which exists unique and complex forwarded and call preference information.

Ideally, standardized definitions would exist for all resources and virtual organizations. Unfortunately, the diversity of real-world applications makes it impractical to create these standardized definitions. As a result directory schemas such as [b-ITU-T X.500] are specifically constructed to be extensible to allow for this customization. Thus, while *inetOrgPerson* is a widely used class to represent people, it is often extended to incorporate custom attributes required for an enterprise.

While resources cannot practically be standardized, it is still important that multimedia conferencing applications have knowledge of the types of resources available so that they can be searched and managed. For example, a video conferencing management application might need to display a list of all video room systems. Similarly, a user might wish to search for conference rooms within a particular building that have conferencing capabilities. A white pages application may allow a user to search for "paediatric doctor on call" without knowing precisely which doctor is on call at any given time. A printed directory application may need to print an entire listing of virtual organizations and people to create a paper phone book.

Applications that access the directory directly, such as those that use H.350, must have direct knowledge of the representation of data in the directory. Specifically, they must be aware of the schemas used to represent people and resources. This is necessary for call signalling and identity management applications. Other applications, such as generic white pages that access inter-domain directories, may not be able to be easily configured for such schema knowledge.

SWIM-D provides a solution for this problem by defining an XHTML interface to a directory proxy that possesses deep knowledge of the directory. Because the directory proxy manages the complexity of the underlying directory, clients and third party systems can access directory data through the directory proxy simply by browsing SWIM-D directory pages, either with a standard HTML client, or a SWIM-D client that conforms to the SWIM-D markup specifications.

7 SWIM-D architecture

SWIM-D provides a mechanism by which organizations can publish directory pages that allow users to recognize, navigate, process and display white pages directory structures.

Figure 1 shows a functional diagram of a complete SWIM-D system.

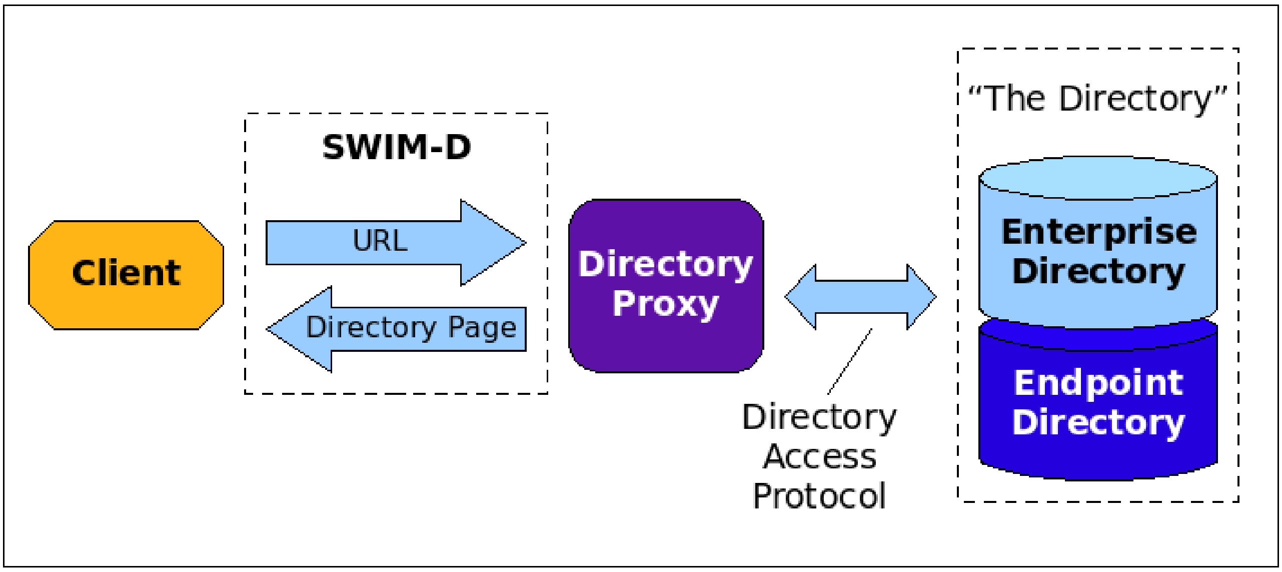

Figure 1 – SWIM-D architectural components

The **directory** in this example comprises the complete set of directory information used to represent people, resources and endpoints. In many cases, it will consist of an extant enterprise directory along with an associated H.350 directory describing all subscribers and their endpoint information. In other cases, it may represent a proprietary directory type with custom attributes, such as a corporate directory with an extended LDAP schema with endpoint information directly in the people or resource objects.

The **directory access protocol** is the protocol used to communicate between the directory and the directory proxy. In the case of H.350, it will be LDAP. However, it could be a structured query language (SQL) call or some other protocol. In most cases, there should be a trust relationship between the directory and the directory proxy. This Recommendation does not specify or even require the use of a directory access protocol, though one will be present in many applications.

The **directory proxy** is a web server that has knowledge, by discovery or configuration, of the directory. It is intended to be straightforward in design, so that it can be easily implemented by an enterprise with knowledge of its own directory structure. It understands the schemas used by the directory to represent people, resources and endpoints and how to search the directory. The directory proxy provides a mapping function between the vocabulary used in the directory schemas and SWIM-D directory pages served out to clients. The directory proxy should authenticate all communications with the directory and with clients. Communications between the directory proxy and the directory and clients should be encrypted. Note that a directory proxy is not required to use a directory access protocol, though it is intended to do so. It could instead serve static SWIM-D directory pages.

SWIM-D is an XHTML over HTTP(S) interface that clients use to access directory information from the directory proxy. SWIM-D describes two elements: **directory pages** and the **SWIM-D URL**. Directory pages are XHTML documents encoded using the metadata described in this Recommendation. These directory pages represent the white pages that users view. The SWIM-D URL is a URL format for directly accessing directory pages and for transmitting information to them to support searching and navigation operations.

A **client** is any system that accesses directory information through the directory proxy using its SWIM-D. Typically, this will be an IP telephone, video terminal, soft client or white pages web application. There is a great degree of variability in the display capabilities of each of these devices. For example, a mobile phone may have an extremely limited display. A video terminal may have a large and colourful display, but may not be well suited for small text. A soft client may be capable of very rich display functions. A white pages application may be a simple web lookup service, or could be an application intended to produce output for eventual printing, such as business cards or telephone books. SWIM-D is intended to provide the directory information and the associated metadata for each of these devices to be able to display the information in the manner most appropriate to the application. Thus, a client can be an HTTP web browser, or an XML-enabled software application capable of interpreting SWIM-D data and metadata.

It is a goal of the overall system design that a single directory proxy will provide information to many different endpoint types, so that a variety of clients from multiple vendors and for multiple applications can be supported from a single interface and with a single system.

Because SWIM-D is a simple HTTP(S) interface, it can be easily implemented on a variety of client types. This architecture simplifies the client's access to the directory because the client does not need to have knowledge of the directory access protocol or the directory itself. This work is off-loaded to the directory proxy.

Note on backward compatibility and accessibility: Because SWIM-D uses XHTML, web browsers that have no knowledge of SWIM-D can still render basic SWIM-D directory pages. This also means that accessibility tools, such as screen readers and speech synthesizers, can operate on SWIM-D directory pages. HTML accessibility guidelines can be supported as well.

8 SWIM-D directory pages

A SWIM-D directory page consists of metadata describing the page and elements that describe the data contained in the page. There are three types of SWIM-D directory pages:

– **MENU directory page**. A navigational page that may contain menu options and a search function. MENU pages can be used to construct controlled navigation options for end users. For example, menu options for "sales", "repair" and "customer service" along with a search function for staff member names could represent a useful navigational page for customers.

– **RESULT directory page**. A summary page returned after a search function is executed. RESULT pages display a list of all the elements that match the search and allow the client to choose among them to further refine the search. For example, searching for "Smith" may return a RESULT page containing all individuals in the directory with last name "Smith". The user can then choose which "Smith" is desired.

– **DETAIL directory page**. A page containing detailed information about a specific directory entry. For example, a DETAIL page might contain all information about an individual, including first name, last name and contact information.

SWIM-D directory pages are intentionally limited in scope to be simple, relatively atomic and single purpose as an aid to support by devices with limited display, memory and processing capabilities and to be backward compatible with non-SWIM-D web browsers. However, advanced applications can be accomplished by powerful clients by combining SWIM-D pages into a single display element. For example, a full display of all users with a last name of "Smith" can be constructed by first obtaining the RESULT page for that search, obtaining the DETAIL page for each entry, and combining that data into a single display. This multiple search capability is non-normative and is up to the implementer as it depends on the capabilities of the client.

SWIM-D directory pages may contain in-line CSS data to represent a style sheet for display purposes. Clients may and generally will provide their own client-side style sheet or equivalent functionality to match SWIM-D elements to the display capabilities and preferences of the clients.

SWIM-D defines the following XHTML *class* values to be used to establish a display priority level for marked up data. These values may be interpreted by clients with limited display capability to determine which elements to display and which to ignore. For example, a client with a single line display may be configured to only display elements whose priority equals 1. Note that more than one element may be of a particular class. It is not the responsibility of the directory page to map priorities to display capabilities. Rather, the directory page indicates the relative importance of items it contains, and the SWIM-D client determines how it will display (or not) these items.

Table 1 shows the XHTML class values that may be used to represent priority. In order to be explicit about describing priority elements, class values shall only be expressed in specific locations within the markup, as described in clauses 8.1, 8.2 and 8.3. Specifically, a SWIM-D element may have a priority attribute only if explicitly stated.

Table 1 – Priority class values

Class value	Purpose
[h.351.priority_1]	Indicates that an element has display priority 1 (the highest priority).
[h.351.priority_2]	Indicates that an element has display priority 2.
[h.351.priority_3]	Indicates that an element has display priority 3.
[h.351.priority_4]	Indicates that an element has display priority 4.
[h.351.priority_5]	Indicates that an element has display priority 5.
[h.351.priority_6]	Indicates that an element has display priority 6.
[h.351.priority_7]	Indicates that an element has display priority 7.
[h.351.priority_8]	Indicates that an element has display priority 8 (the lowest priority).

SWIM-D directory pages contain the following metadata elements:

```
    <meta name="h.351_version" content="1.0"/>
```

The `h.351_version` property identifies that the directory page is an H.351 page and the version of H.351 supported is version 1.0.

8.1 MENU directory pages

A MENU directory page contains metadata to indicate that it is a SWIM-D MENU directory page. It also contains SEARCH and MENU_LINK elements.

A start directory page is a special case of a MENU directory page that serves as the entry point for SWIM-D browsing. The URL of the start directory page is the only URL that has to be known by the client in order to use the SWIM-D service. It has to be configured on each client and represents a "home" page for that CLIENT. Start page is a non-normative term and is only meant as a convenience and convention.

8.1.1 Metadata

A MENU directory page contains the following metadata elements:

```
<meta name="h.351" content="menu">
```

The `h.351v1.0` property identifies the page as being of type H.351 SWIM-D and the content indicates that it is a MENU directory page.

8.1.2 Elements

A MENU directory page contains one SEARCH, or one MENU_LINKS element, or one of both. A CLIENT uses the directory page elements to render a search mask or a navigational menu, respectively.

8.1.2.1 SEARCH elements

A SEARCH element is an XHTML *forms* construct of one or more XHTML form controls that can be used by the client to create a directory search mask. Each form control has a NAME, a LABEL and a VALUE.

– **NAME**: The control name of type TEXT signified by the XHTML *name* element. The name shall be a string that is equivalent to the search variable name that the server expects for a query operation.

– **LABEL**: The LABEL is the XHTML *label* element and is used by the client to label the search fields for display purposes. LABEL is used with *for* and *id* elements to associate the label with a particular form NAME as specified in [W3C XHTML 1.1].

– **VALUE**: The VALUE is the text entered by the user, and is not included in the markup of a SEARCH element, but is used to construct the SEARCH MASK in the SWIM-D directory page URL as described in clause 9.

SEARCH elements may have a priority class assigned. Priority is assigned to individual form controls, not to the entire form.

The following values of *class* are defined for search elements:

***class* value**	**Purpose**
[h.351.search]	Indicates that a form is a search element.

8.1.2.2 MENU_LINK elements

A MENU_LINK element is an *unordered list* of one or more MENU_LINK definitions that include hyperlinks to directory pages. A MENU_LINK definition consists of exactly one link label and URL pair.

MENU_LINK definitions may have a priority class assigned.

The following values of *class* are defined for MENU_LINK elements:

***class* value**	**Purpose**
[h.351.menu_link]	Indicates that a label/value pair is a menu link element.
[h.351.menu_link_label]	Indicates a label that can be used for user-friendly display purposes.
[h.351.menu_link_url]	Indicates a value for the entry detail. Typically, a SWIM-D URL.

Example MENU directory page

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

    <head>
        <title>SWIM-D Menu Directory Page Example</title>
        <meta http-equiv="content-type" content="text/html;charset=utf-8" />
        <meta http-equiv="Content-Style-Type" content="text/css" />
        <meta name="h.351_version" content="1.0"/>
        <meta name="h.351" content="menu"/>
    </head>

    <body>
        <p>Customer Service Directory</p>

        <ul>
            <li class="h.351.menu_link h.351.priority_1" >
                <span class="h.351.menu_link_label">
                  Sales Department</span>:&nbsp;
                <span class="h.351.menu_link_url">
                  <a href="https:dir.example.com/sales.html">
                  click here for Sales</a></span>
            </li>
            <li class="h.351.menu_link h.351.priority_1">
                <span class="h.351.menu_link_label">
                  Repair Department</span>:&nbsp;
                <span class="h.351.menu_link_url">
```

```
                  <a href="https:dir.example.com/repair.html">
                  click here for Repair</a></span>
            </li>
        </ul>

        <p>Search for an Individual</p>

        <form class="h.351.search" action="http://www.example.com/lookup.php"
          method="get">
            <p>
                <label for="lastname">Last Name</label>
                <input class="h.351.priority_2" type="text" name="lastname"
                  value="" id="lastname"/> <br />
                <label for="firstname">First Name</label>
                <input class="h.351.priority_3" type="text" name="firstname"
                  value="" id="firstname"/> <br />
                <input type="submit" />
            </p>
        </form>

    </body>
</html>
```

8.2 RESULT directory pages

A RESULT directory page contains a table of the search results, allowing the user to further refine the search.

8.2.1 Metadata

A MENU directory page contains the following metadata elements:

```
     <meta name="h.351" content="result"/>
     <meta name="h.351_result_totalnum" content="[integer]"/>
     <meta name="h.351_result_startnum" content="[integer]"/>
     <meta name="h.351_result_listnum" content="[integer]"/>
```

The `h.351` property identifies the page as being of type H.351 SWIM-D and the content indicates that it is a RESULT directory page.

The `h.351_result_totalnum` property identifies that the search resulted in [integer] results, though not all of them are necessarily contained on the page.

The `h.351_result_startnum` property states that the entries contained on the page begin at number [integer] of the entire set of results.

The `h.351_result_listnum` property states that the page contains [integer] results.

Taken together, the totalnum, startnum and listnum elements can be used to create both client side and server side navigational aides for content browsing, such as the following page display fragment:

<< 1 2 3 **4** 5 >>

8.2.2 RESULT element

A RESULT element is a *table* representing the result of a directory search operation. A directory entry line contains summary data and a hyperlink pointing to the entry's detail directory page. The number of columns and rows returned is determined by arguments in the search URL. Elements marked with `<th>` and `</th>` tags identify column labels for non-HTML clients.

Priority may be assigned to table columns.

The following values of *class* are defined for entry detail elements:

***class* value**	**Purpose**
[h.351.detail_url]	Indicates that an element is a SWIM-D detail URL.

Example results directory page

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

    <head>
        <title>SWIM-D Results Directory Page Example</title>
        <meta http-equiv="content-type" content="text/html;charset=utf-8" />
        <meta http-equiv="Content-Style-Type" content="text/css" />
        <meta name="h.351_version" content="1.0"/>
        <meta name="h.351" content="result"/>
        <meta name="h.351_result_totalnum" content="10"/>
        <meta name="h.351_result_startnum" content="1"/>
        <meta name="h.351_result_listnum" content="3"/>
    </head>

    <body>
        <p>Search Results</p>

        <table>
            <tr>
                <th class="h.351.priority_2">Last Name</th>
                <th class="h.351.priority_1">First Name</th>
                <th class="h.351.priority_1">More Info</th>
            </tr>
            <tr>
                <td>Smith</td>
                <td>John</td>
                <td><a class="h.351.detail_url"
                     href="https://dir.example.com/?a=details&amp;entryid=2001">
                     show details</a></td>
            </tr>
            <tr>
                <td>Smith</td>
                <td>Kimberly</td>
                <td><a class="h.351.detail_url"
                     href="https://dir.example.com/?a=details&amp;entryid=2002">
                     show details</a></td>
            </tr>
            <tr>
                <td>Smith</td>
                <td>Liam</td>
                <td><a class="h.351.detail_url"
                     href="https://dir.example.com/people/smith_liam.html">
                     show details</a></td>
            </tr>
        </table>

    </body>
</html>
```

8.3 DETAIL directory pages

A DETAIL directory page displays full details of a directory entry.

8.3.1 Metadata

A MENU directory page contains the following metadata elements:

```
<meta name="h.351" content="detail"/>
```

8.3.2 DETAIL element

An *unordered list* of one or more directory entry elements. An entry detail element contains exactly one label and value pair.

Priority may be assigned to list items.

The following values of *class* are defined for entry detail elements:

***class* value**	**Purpose**
[h.351.entry_detail]	Indicates that a label/value pair is an entry detail element.
[h.351.entry_detail_label]	Indicates a label that can be used for user-friendly display purposes.
[h.351.entry_detail_value]	Indicates a value of the entry detail. Typically, a search result.

NOTE – In the following example, the text:

```
          :&nbsp;
```

is used to denote a separator to display, for example:

```
          Last Name: Smith
```

This Recommendation does not define XHTML markup for display purposes. Implementers could choose, for example:

```
     <span class="separator">;&nbsp;</span>
```

to generalize the markup for more advanced display options, such as supporting international customization.

Example DETAIL directory page:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

    <head>
        <title>SWIM-D Detail Directory Page Example</title>
        <meta http-equiv="content-type" content="text/html;charset=utf-8" />
        <meta http-equiv="Content-Style-Type" content="text/css" />
        <meta name="h.351_version" content="1.0"/>
        <meta name="h.351" content="detail"/>
    </head>

    <body>
        <p>Contact Details</p>

        <ul>
            <li class="h.351.entry_detail h.351.priority_2">
                <span class="h.351.entry_detail_label">Last Name</span>:&nbsp;
                <span class="h.351.entry_detail_value">Smith</span>
            </li>
            <li class="h.351.entry_detail h.351.priority_2">
                <span class="h.351.entry_detail_label">FirstName</span>
                   :&nbsp;
                <span class="h.351.entry_detail_value">John</span>
            </li>
            <li class="h.351.entry_detail h.351.priority_1">
                <span class="h.351.entry_detail_label">Address</span>:&nbsp;
                <span class="h.351.entry_detail_value">
                    <a href="sip:John_Smith@example.com">
                       sip:John_Smith@example.com</a></span>
            </li>
        </ul>

    </body>
</html>
```

9 SWIM-D URLs

A directory page is uniquely identified by its URL, and a client navigates through directory pages using URL hyperlinks included in the pages. The general syntax of a directory page URL follows the HTTP URI scheme as defined in [IETF RFC 2616] and the HTTPS URI scheme as defined in [IETF RFC 2818]. Both absolute and relative URLs shall be supported by clients and directory proxies.

There are three types of SWIM-D URLs:

– **MENU directory page URL**. Instructs the directory proxy to return a MENU directory page.

– **SEARCH directory page URL**. Instructs the directory proxy to return a RESULT directory page.

– **DETAIL directory page URL**. Instructs the directory proxy to return a DETAIL directory page.

9.1 MENU directory page URLs

The format of a **MENU directory page URL** is defined as:

```
  http(s): <Directory Proxy hostname/ip> <Directory Page path>
```

– **<Directory Proxy hostname/ip>** The fully qualified hostname of the directory proxy. In case the directory proxy has no DNS entry, an IPv4 or IPv6 address may be used.

– **<Directory Page path>** This path uniquely identifies a MENU directory page. It consists of path fragments separated by the "/" character similar to a file system path. The directory page path can be an empty string.

9.2 SEARCH directory page URLs

The format of a **SEARCH directory page URL** is defined as:

```
   http(s): <Directory Proxy hostname/ip> <Directory Page path> ? a=search
            [ & <search name>=<search value> ]*
            [ & maxcolumns=<max number of returned columns> ]
            [ & maxrows=<max number of returned rows> ]
            [ & startnum=<result set index> ]
```

– **<search name>, <search value>**

The <search name> equals a NAME element from a menu directory page form element. <search value> is the value entered by a user in the directory search mask. Required.

- **<max number of returned columns>**

  This value specifies the maximum number of columns that should be returned by the directory proxy.

- **<max number of returned rows>**

  This value specifies the maximum number of rows, representing individual directory search results, that should be returned by the directory proxy.

- **<result set index>**

  A client can use this parameter to tell the directory proxy the desired result set index. This allows to page through a set of resulting directory entries.

9.3 DETAIL directory page URLs

The format of a **DETAIL directory page URL** is defined as:

```
http(s): <Directory Proxy hostname/ip> <Directory Page path>
         [filename] &or ? [selection_filter]
```

- <**filename**>

  The name of the detail directory page, such as `smith_liam.html`. More typically, a selection filter will be used instead of a specific filename.

- <**selection filter**>

  This optional string is defined and generated by the directory proxy and is used to pass specific variable names between the client and directory proxy. See the results directory page example for an example of this syntax.

10 General considerations

10.1 Usage of HTTP

A directory proxy shall support both HTTP and HTTP over TLS (HTTPS) following the procedures in [IETF RFC 2818]. Furthermore, a directory proxy shall be able to enforce HTTPS even if a client tries to connect via HTTP. A client should implement HTTPS as specified in [IETF RFC 2818] and use it as the default transport protocol. A client may use HTTP if client resources do not allow for TLS encryption.

10.2 Authentication and authorization

HTTP digest access authentication as defined in [IETF RFC 2617] is used for client authentication. The directory proxy authorizes a client based on the credentials used during the digest authentication process

10.3 Directory browsing

To begin the browsing process, a client issues an HTTP GET request with the SWIM-D start directory page URL. A client may use another SWIM-D directory page URL stored locally on the client. This URL can be configured through client provisioning or saved as a bookmark hyperlink by the user. Once the client receives a valid directory page, it uses the page elements to render the directory user interface.

If the returned directory page includes a SEARCH element, a client with a user interface should render a directory search mask including at least the default search field definition. This search mask MUST allow a user to enter search values according to the SEARCH definitions and to perform a directory search operation.

If a MENU_LINKS element is returned, a client with a user interface should present the included hyperlinks to the user and offer a way to select one of the hyperlinks. Selecting a hyperlink shall result in a new SWIM-D request using the selected hyperlink. The client shall execute the rules specified in this clause once a new directory page is returned.

10.4 Directory searching

A client uses the SEARCH directory page element to render a directory search mask as defined in clause 8.1.2.1. A directory search operation is performed by issuing a new SWIM-D request with a SEARCH directory page URL. This URL shall comply with the SEARCH directory page URL format and is constructed by the following rule:

1) Take the current SWIM-D URL and remove any query terms.

2) Add the query term `a=search`.

3) If a user has entered values in the directory search mask, add a query term `<search name>=<search value>` for each search value.

4) Optionally, add `maxcolumns=<max number of returned results>`, `maxrows=<max number of returned rows>` and/or `startnum=<result set index>`.

An example SEARCH directory page URL looks like:

```
https://dir.example.com/hr?a=search&fname=peter&sname=smith
```

Note that this process is an implementation of XHTML forms as described in [W3C XHTML 1.1]. Clients may use HTTP GET or HTTP POST methods, but generally should use GET for read-only applications.

After issuance of a directory search request to a directory proxy, the directory proxy must return a corresponding result directory page.

10.5 Extensibility

XHTML is designed to be stably extensible and SWIM-D supports this extensibility in the following ways:

– Implementers may create their own class definitions for use in SWIM-D directory pages. The name space for these extensions shall not conflict with classes already defined for SWIM-D in this Recommendation. As is the case for HTML browsers, non-HTML SWIM-D clients should ignore class definitions that they do not recognize. Thus, proprietary extensions should be ignored by, but not be broken by, unrecognized attributes and attribute values.

– Implementers may create their own `<META>` tags and assign values to them. The name space for these `<META>` elements shall not conflict with elements already defined for SWIM-D in this Recommendation.

– Implementers may not alter the URL syntax of SWIM-D.

Bibliography

[b-ITU-T E.115]	Recommendation ITU-T E.115 (2007), *Computerized directory assistance*.
[b-ITU-T H.323]	Recommendation ITU-T H.323 (2006), *Packet-based multimedia communications systems*.
[b-ITU-T H.35x]	Recommendation ITU-T H.35x-series (2003), *Directory services architecture for multimedia conferencing*.
[b-ITU-T X.500]	Recommendation ITU-T X.500 (2005) | ISO/IEC 9594-1:2005, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services*.
[b-IETF RFC 3261]	IETF RFC 3261 (2002), *SIP: Session Initiation Protocol*. <http://www.ietf.org/rfc/rfc3261.txt>
[b-IETF RFC 3377]	IETF RFC 3377 (2002), *Lightweight Directory Access Protocol (v3): Technical Specification*. <http://www.ietf.org/rfc/rfc3377.txt>
[b-IETF RFC 4510]	IETF RFC 4510 (2006), *Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map*. <http://www.ietf.org/rfc/rfc4510.txt>
[b-W3C HTML 4.01]	W3C Recommendation (1999), *HTML 4.01 Specification*. <http://www.w3.org/TR/REC-html40/>
[b-W3C XML 1.0]	W3C Recommendation (2008), *Extensible Markup Language (XML) 1.0.* <http://www.w3.org/TR/2008/PER-xml-20080205/>

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2009