The RX Document
Version 1.0
X11 Release 6.4

Arnaud Le Hors
lehors@x.org
X Consortium, Inc.

Abstract

This document describes the RX MIME type and how it can be used to provide a means to execute remote applications, such as X Window System clients, from a World Wide Web browser. To achieve this, the RX document must convey enough information for an application to get everything it needs to connect to the various resources available in the user's environment and display a graphic user interface, possibly passing through some security firewall.

1. Introduction

The RX MIME type [4] document is designed to provide a means to execute remote applications, such as X Window System clients, from a World Wide Web browser. However, the document itself is not sufficient. To be of any use. it requires the browser to use the information it contains and to react accordingly. The general model is that the RX document conveys to the browser the list of services the application needs to run. In reaction the browser sets up the user environment for the application to run, and starts the application by feeding the web server back the relevant information. The RX document can list both required and optional services, leaving the browser to decide, possibly based on user preferences, which services are available and/or preferred.

This document describes the RX MIME type and how the browser is expected to start the application by running a CGI script or equivalent [6].

2. Notational Conventions and Generic Grammar

All of the mechanisms specified in this document are described in both prose and an augmented Backus-Naur Form (BNF). Readers need to be familiar with the notation in order to understand this specification. The augmented BNF includes the following constructs:

name = definition

> The name of a rule is simply the name itself (without any enclosing “<“ and “>”) and is separated from its definition by the equal character “=”. Whitespace is only significant in that indentation of continuation lines is used to indicate a rule definition that spans more than one line. Angle brackets are used within definitions whenever their presence will facilitate discerning the use of rule names.

“literal”

> Quotation marks surround literal text. Unless stated otherwise, the text is case-insensitive.

rule1 | rule2

> Elements separated by a bar (“|”) are alternatives, e.g., “yes | no” will accept yes or no.

(rule1 rule2)

> Elements enclosed in parentheses are treated as a single element. Thus, “(elem (foo | bar) elem)” allows the token sequences “elem foo elem” and “elem bar elem”.

*rule

> The character “*” preceding an element indicates repetition. The full form is “<n>*<m>element” indicating at least <n> and at most <m> occurrences of element. Default values are 0 and infinity so that “*(element)” allows any number, including zero; “1*element” requires at least one; and “1*2element” allows one or two.

[rule]

Square brackets enclose optional elements; “[foo bar]” is equivalent to “*1(foo bar)”.

2.1 Basic Rules

The following rules are used throughout this specification to describe basic parsing constructs.

OCTET = <any 8-bit sequence of data>

ALPHA = <any US-ASCII letter “A”..”Z” and “a”..”z”>

DIGIT = <any US-ASCII digit “0”..”9”>

CHAR = <any character>

CTL = <any US-ASCII control character
(octets 0 - 31) and DEL (127)>

CR = <US-ASCII CR, carriage return (13)>

LF = <US-ASCII LF, linefeed (10)>

SP = <US-ASCII SP, space (32)>

HT = <US-ASCII HT, horizontal-tab (9)>

SPECIAL = “;” | “?” | “,” | “=” | “<“ | “>” | SP | HT | CR | LF

STRING = 1*<any CHAR except CTLs and SPECIALs>

TEXT = <any OCTET except CTLs, but including CR and LF>

3. The RX MIME type

3.1 General form

The general form of an RX document is a list of HTML PARAM elements, as drafted by the WWW Consortium [5] in its simplest form:

“<PARAM NAME=” name “VALUE=” value “>”

One advantage of this syntax is to provide HTML authors with the possibility to easily move elements from the RX document to the HTML document itself, when the OBJECT tag becomes standard and is fully supported by web browsers. The list of possible parameters is defined in the following section.

In addition the standard HTML comment element, which is of the form:

“<!--” TEXT “-->”

is supported.

3.2 Parameters

First, an RX document can specify the RX version number with the VERSION parameter which can have the following value (in augmented BNF):

version = 1*DIGIT "." 1*DIGIT

This allows the document to specify a major and minor numbers. When not specified the default value for this parameter is 1.0. When specified this must appear first in the document.

Then, an RX document can contain any of the parameters described below, in any order:

ACTION

The URL to fetch to initiate the remote execution. It is equivalent to the DATA attribute of the OBJECT element and the SRC parameter of the Netscape™ EMBED element.

REQUIRED-SERVICES

This indicates the list of services the application needs to run. This list can contain one or more of UI, PRINT as described below.

UI

This indicates the list of User Interface protocols the application supports, in order of preference (from the most to the least preferred).

PRINT

This indicates the list of printing protocols the application support, in order of preference (from the most to the least preferred).

WIDTH and HEIGHT

This allows the HTML author to specify the default geometry of the application primary window.

EMBEDDED

This indicates whether the application is to be embedded in the browser or not. If not specified in the RX document, the default value is YES, unless overridden by browser or user settings.

AUTO-START

This specifies whether the application should be launched immediately upon retrieval of the RX document or only on user demand. For instance, this could be on a user click on the embedded region, or whatever other user interface element the browser wants to provide for this. If not specified in the RX document, the default value is YES, unless overridden by browser or user settings.

APP-GROUP

This specifies the logical application group to which the application is related. The intent of this attribute is to allow several remote applications to be considered as a single logical application or, on the contrary, to separate various applications from each other. If not specified in the RX document, the application is considered to be alone within its own logical group.

The possible values for each of these parameters is respectively defined as follows (in augmented BNF):

action = <script URI>
required-services = service *(“,” service)
service = “UI” | “PRINT”
ui = protocols
protocols = protocol *(“,” protocol)
protocol = STRING
print = protocols
width = 1*DIGIT
height = 1*DIGIT
embedded = “YES” | “NO”
auto-start = “YES” | “NO”
app-group = STRING

In addition to this exhaustive list of parameters, an RX document can also contain parameters only relevant to a particular protocol. The names of these parameters must be prefixed by the related protocol name and the value can be anything. There may be as many such parameters as desired. These parameters are defined as follows:

param-name = protocol “-” STRING
param-value = STRING

Every parameter can also be specified in the HTML document as an attribute of the OBJECT or EMBED element when one is used. When this is the case and a parameter is specified both in HTML and RX, the HTML instance has precedence over the RX one. The only exception to this rule is the VERSION parameter which only affects the set of parameters to which it is attached, whether this is in HTML or RX.

3.3 Returned Parameters

Once the browser has read the RX document and performed any necessary initialization, in response it should start the application at the appropriate time (see auto-start) by fetching the action URI, via an HTTP GET request [3], with the following parameters list:

```
return-parameters       = *( "?" return-parameter )
   return-parameter     = ui-return | print-return
                          | width-return | height-return | embedded-return
                          | extra-param-return
ui-return               = "UI=" ui-url
print-return            = "PRINT=" print-url
width-return            = "WIDTH=" 1*DIGIT
height-return           = "HEIGHT=" 1*DIGIT
embedded-return         = "EMBEDDED=" embedded
extra-param-return      = protocol "-" STRING "=" STRING
```

For each requested service specified in the RX document the web browser can respond with the corresponding return parameter. If any of the requested services is not listed in the returned parameters, it should be assumed that the service is unavailable.

The action URI, which should be a CGI script or equivalent, must not only start the application with the given parameters, it is also expected to produce a valid document of type "text/plain". The first line of this document must contain an error code, either 0 for success or non zero for failure. The non existence of this line should be considered as a sign of failure. The rest of the document can then contain some error messages to be displayed by the browser.

4. How the RX document will be used in the X Window System

4.1 Parameters

In X for the UI and PRINT services respectively the protocols X and XPRINT are supported and are then valid protocol names.

Also the following additional parameters are recognized:

X-UI-LBX, X-PRINT-LBX

These parameters indicate whether or not the application is LBX capable. If not specified in the RX document, the default value is NO, unless overridden by browser or user settings.

X-UI-INPUT-METHOD

This indicates that the application can make use of an input method server. If not specified in the RX document, the default value is NO, unless overridden by browser or user settings.

X-AUTH

This parameter specifies a default list of authentication mechanisms, in order of preference (from the most to the least preferred), applying to all services.

X-UI-AUTH, X-PRINT-AUTH, X-UI-LBX-AUTH, and X-PRINT-LBX-AUTH

These parameters specify a list of authentication mechanisms, in order of preference (from the most to the least preferred), applying to one service in particular, that is X, XPRINT, LBX for X, and LBX for XPRINT respectively. They have precedence over the global X-AUTH parameter when both are specified.

The X-UI-LBX, X-PRINT-LBX parameters can have the following value:

x-lbx	= "YES" | "NO"

The X-UI-INPUT-METHOD parameter can have the following value:

x-ui-input-method	= ("YES" [";" URL]) | "NO"

This specifies whether an input method server is available or not, and if one is its location can be specified as an URL.

Each of the X-UI-AUTH, X-PRINT-AUTH, X-UI-LBX-AUTH, and X-PRINT-LBX-AUTH parameters have the following value:

xauths	= xauth *("," xauth)
xauth	= xauth-name [":" xauth-data]
xauth-name	= 1*<any CHAR except CTLs, SPECIALs and ":">
xauth-data	= STRING

The authentication mechanism name can be for instance MIT-MAGIC-COOKIE-1 or MIT-KERBEROS-5 while the authentication data would be, respectively, nothing or a Kerberos principal and realm; for instance "webserver@mycorp.com".

4.2 Returned parameters

In the GET request, the returned URLs for the UI and PRINT services, that is for the X and XPRINT protocols, are of the form:

ui-url	= "x11:" xdisplay [";" auth]
xdisplay	= display | decnet-display
display	= [transport "/"] host ":" display-num ["." screen-num]
transport	= "local" | "tcp" | "decnet"
decnet-display[1]	= host "::" display-num ["." screen-num]
host	= <A legal Internet host domain name or IP address (in dotted-decimal form), as defined by Section 2.1 of RFC 1123>

1. this syntax is supported for compatibility with Xlib and is equivalent to the general display specification with decnet as the transport.

display-num = 1*DIGIT
screen-num = 1*DIGIT
auth = “auth=” xauth[1]
print-url = “xprint:” xprinter [“;” auth]
xprinter = [printer-name “@”][transport “/”] host “:” display-num
printer-name = 1*<any CHAR except CTLs, SPECIALs and “@”>

And in response to the X-UI-LBX, X-PRINT-LBX parameters, the following is given as part of the GET request:

x-ui-lbx-return = “X-UI-LBX=” (“YES” [“;” auth]) | “NO”
x-print-lbx-return = “X-PRINT-LBX=” (“YES” [“;” auth]) | “NO”

4.3 Example

An HTML document specifying an RX MIME-type document using the OBJECT tag would look like:

```
<HTML>
...
<OBJECT data=http://www.domain.com/CalendarTool.rx
        type=application/x-rx
        width=500 height=400>
<IMG SRC=http://www.domain.com/CalendarTool.gif ALT=CalendarTool>
</OBJECT>
...
</HTML>
```

With the Netscape™ EMBED tag it would look like:[2]

```
<HTML>
...
<EMBED SRC=http://www.domain.com/CalendarTool.rx width=500 height=400>
...
</HTML>
```

The document SimCity.rx could contain the following:

```
<PARAM Name=VERSION Value=1.0>
<PARAM Name=ACTION Value=http://www.domain.com/CalendarTool.pl>
<PARAM Name=REQUIRED-SERVICES Value=UI>
<PARAM Name=UI VALUE=X>
```

1. xauth is defined further.
2. One cannot specify an alternative such as, an image, within the EMBED tag.

```
<PARAM Name=X-UI-LBX Value=YES>
<PARAM Name=X-AUTH Value=MIT-MAGIC-COOKIE-1>
<PARAM Name=EMBEDDED Value=YES>
<PARAM Name=APP-GROUP Value=CalendarToolAppGroup1>
```

From which the browser would reply by a GET of the following URI:

```
http://www.domain.com/CalendarTool.pl
      ?UI=x11:myhost.mydomain.org:0;auth=MIT-MAGIC-COOKIE-1:044B3244D
      ?WIDTH=500?HEIGHT=400?EMBEDDED=YES
      ?X-UI-LBX=YES;auth=MIT-MAGIC-COOKIE-1:1A7C4C1F312B3
```

Note that the URI is really made of a single line and is only presented on several lines in this document to make it easier to read.

5. References

[1] T. Berners-Lee, "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", RFC 1630, CERN, June 1994 (http://ds.internic.net/rfc/rfc1630.txt).

[2] T. Berners-Lee, L. Masinter, and M. McCahill., "Uniform Resource Locators (URL)", RFC 1738, CERN, Xerox Corporation, University of Minnesota, December 1994 (http://ds.internic.net/rfc/rfc1738.txt).

[3] T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext Transfer Protocol -- HTTP/1.0", May 1996 (http://www.ics.uci.edu/pub/ietf/http/rfc1945.html).

[4] N. Borenstein and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September 1993 (http://ds.internic.net/rfc/rfc1521.txt).

[5] Dave Raggett, "Inserting objects into HTML", W3C Working Draft 22-Apr-96 (http://www.w3.org/pub/WWW/TR/WD-object-960422.html).

[6] David Robinson, "The WWW Common Gateway Interface Version 1.1", University of Cambridge UK, 15 February 1996 (http://www.ast.cam.ac.uk/~drtr/cgi-spec.html).