XML-F

Preliminary Specification

Rev. 0.1

August, 1998

Notice: This is a “preliminary” specification presented as a basis for discussion regarding a common application interface. This is not an approved or implemented standard and the specifications herein are likely to change.

The Leader in Integrated Fax

The XML-F Interface DTDs

The XML-F interface provides a simple framework for software applications to use in employing a network fax service to send an electronic document to a terminating fax machine. The XML-F interface supports three basic features:

(1) Submit a Fax for Transmission

(2) Get Status of a Fax Transmission

(3) Cancel a current Fax Transmission or Request

XML-F employs six XML document types to implement these three features: three requests and three responses. Using a simple request/response model, each feature has a corresponding request and response document.

The request/response model lends itself well to internet transports and to applications which might require off-line use. All data is formatted as text and binary data is encoded using Base64 encoding.

XML-F can be coupled with internet transports such as TCP/IP and HTTP and security mechanisms such as SSL to provide secure transactions over public networks. XML-F does not imply or require any particular transport, however.

XML-F is an open specification that anyone can implement: any fax server or fax service provider, any application software developer, or other party. It is recommended that any public implementation of a server representing this interface fully implement all of the interface, while client systems might opt to implement only those portions which are relevant to the application.

Philosophy of Design

The primary design philosophy behind XML-F is that the resultant specification be *simple*. Simplicity is at the heart of any commonly-adopted protocol or interface and allows more people to understand and get productive with the interface sooner. The XML-F interface is simple enough for application developers of all levels to be able to use with only minimum investment. Since XML-F is based on XML, a plethora of tutorials, XML editors, and tools are available to expedite one's familiarity with the underlying language.

In order to drive participation in refining XML-F and to encourage early implementations, XML-F has been defined very tightly. Specifically, there are no facilities for user-logon, authentication, delivery or statusing of received faxes, user-logs, capabilities exchange or system configuration. Most of these facilities are rarely

required of an application integration to fax, others, while potentially useful, would hinder the discussion around the basic premise that XML-F is trying to advance at this time— merely by adding too much complexity to the initial specification.

Secondly, XML-F describes an interface between an application and fax service where there is a *minimized contract* between the components. That is, there is little knowledge that the application is required to have about the fax service (and vice versa) – only that which is in the XML-F specification can and should be assumed by the application and fax service.

For example, the application in XML-F may request that a service either provide a coversheet service or not – there is no facility to "select" a coversheet resident on the server. Such a facility would require knowledge of the resources of the server which are not available to the application.

Thirdly, XML-F seeks to be *consistent* both internally and with respect to the specifications and requirements recommended by the designers of XML.

The design of XML-F itself is entirely synchronous: the client application makes a request of the server, and the server responds only to one of these requests. An asynchronous method can be constructed by using the server's ability to send email, but it is the responsibility of the client to rendezvous (somehow) with these emailed responses.

XML-F Application Domain

The following figure depicts a typical exchange between an application and a fax server using XML-F. Most applications merely create a document in some standard printable

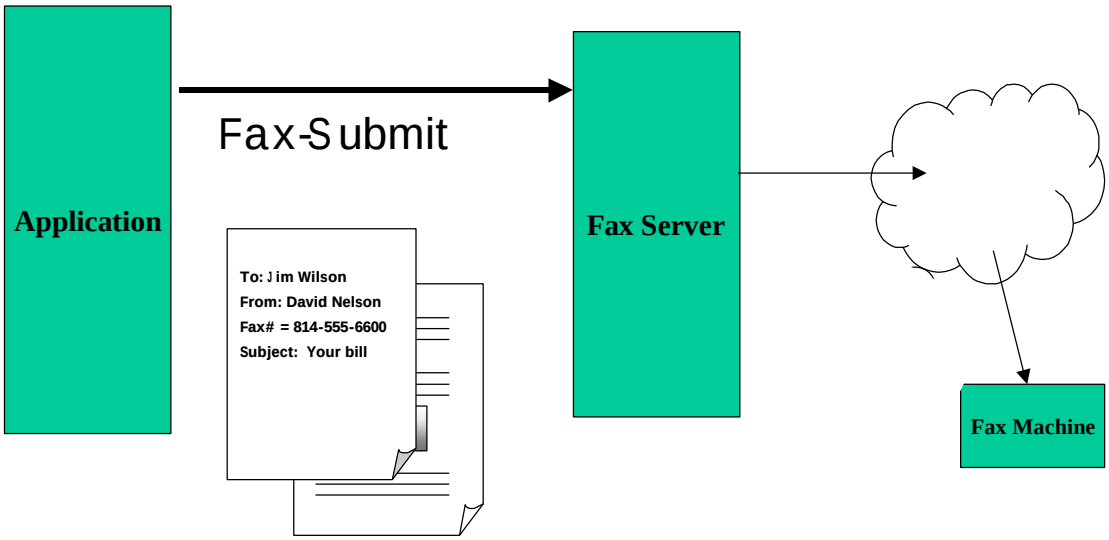

Figure 1 Typical Fax Integration

format (either text, PCL™, or PostScript™), and create a separate list of fax-sending parameters which identify who the fax is going to, what their fax number is, who is sending it, and some other standard cover sheet information.

This information is somehow transported to the fax server (either through an API, or by dropping the file into a specific network location polled by the fax server) and the fax server is responsible for getting the fax sent to the designated fax machine. If something goes wrong with the fax transmission, the fax server will respond to the user via email or to the sending application through the API.

Some integrations require tighter response and abilities by the application, that is, the application *itself* needs to know and record the completion and status of the sent fax and update some internal database of that event. In the XML-F model, the application requests status from the fax server on the particular fax-submittal and the fax server responds in kind with a short or detailed fax status report (also delivered as an XML document).

In some cases, the application can provide its user with the ability to cancel an in-process transaction, and this capability is also frequently provided in an application-to-fax integration.

The following diagram, outlines the six transfers that occur in a fully implemented XML-F system:

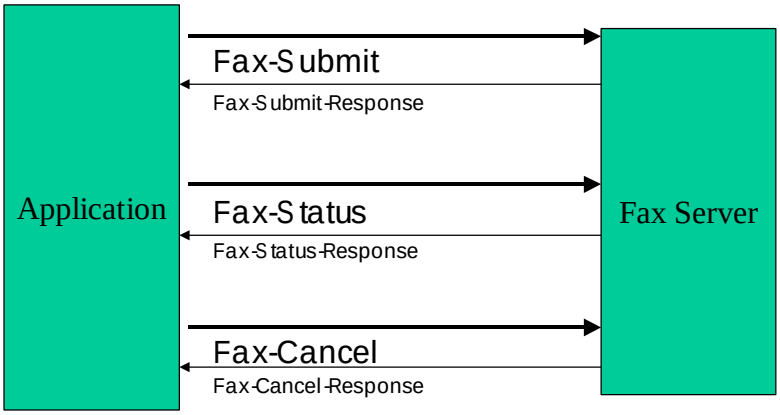

Figure 2 Six XML-F Documents

Transports and APIs

XML-F is transport independent. In the same way that HTML files can be retrieved over HTTP or emailed over SMTP/MIME or shared across file systems, so too, is XML-F. The structure of XML-F lends itself to either synchronous or asynchronous transport models although a synchronous model is assumed.

A common API for XML-F would be highly desirable and is being considered as an extension to this work. This API must be contemplated in light of recent efforts to standardize the programming interfaces to XML base objects in a common object model and in the work by Microsoft, DataChannel and others to standardize brokering services of XML objects.

The Fax-Submit Document

The `fax-submit` document describes the elements necessary to send a fax on an XML-F compliant system. The `fax-submit` document maintains the following structure:

```
fax-submit
      account                              (required)
            id                             (required)
            subid                          (zero or one)
            mailaddress                    (required)
      recipient                            (one or more)
            personal-name                  (zero or one)
            company-name                   (zero or one)
            fax-number                     (required)
            voice-number                   (zero or one)
      sender                               (required)
            personal-name                  (required)
            company-name                   (zero or one)
            fax-number                     (zero or one)
            voice-number                   (zero or one)
      control                              (zero or one)
            application-reference          (zero or one)
            resolution                     (zero or one)
            priority                       (low, "normal", high)
            coversheet                     (Yes or No)
            email-notification             ("none", on-failure, etc.)
      content                              (one or more)
            subject                        (zero or one)
            body [URL or BodyFILE]         (one or more)
      command-reference                    (zero or one)
```

The `account` contains the information necessary to identify the user of the service. Authentication is left to the transport and implementation, the account structure contains an `id` intended to identify the billing entity and a *sub-id* to be used for grouping of departments or users within the billing entity. The `account` also requires a `mailaddress` which is an email channel available for the fax service to use for error or mandatory messages.

The recipient defines one or more people for whom this fax transmission is ultimately intended. A fax submission must have at least one recipient, however, the recipient need only have fax-number defined.

The sender is a required entity with at least the personal-name defined for the sender.

The control entity is optional and provides controls for resolution and priority. The specification allows for a coversheet control to specify that a coversheet is required on the service side— not a selection of a particular coversheet. An application-reference tag is defined to permit the submitting application to provide an arbitrary application-specific reference to this fax request. This reference tag may be used to request status information on this fax submission.

The content element contains a subject (optional) and one or more body elements which can either be URLs accessible from the fax service, or in-line bodyfiles which have an associated mime-type (RFC 2046) and encoding-scheme (base64 or none).

Finally, in XML-F, all requests carry a command-reference which is intended for the sending system to uniquely identify *this* particular request. This is useful for debugging and matching up particular responses with particular requests when the system is used asynchronously.

Upon successful receipt of a valid fax-submit, a fax-submit-response will be returned.

```
<!-------------------------------------------------------------->
<!-- XML-F Fax-Submit DTD                                      -->
<!--                                                           -->
<!-- Used for submitting a fax to an XML-F-conformant server. -->
<!-- Server may respond with a fax-submit-response            -->
<!--                                                           -->
<!-------------------------------------------------------------->

<!ELEMENT fax-submit                  (account,
                                       recipient+,
                                       sender,
                                       control?,
                                       content,
                                       command-reference?)>
<!ELEMENT account                     (id,
                                       subid?,
                                       mailaddress)>
<!ELEMENT id                          (#PCDATA)>
<!ELEMENT subid                       (#PCDATA)>
<!ELEMENT mailaddress                 (#PCDATA)>
<!ELEMENT recipient                   (personal-name?,
                                       company-name?,
                                       fax-number,
                                       voice-number?)>
<!ELEMENT personal-name               (#PCDATA)>
<!ELEMENT company-name                (#PCDATA)>
<!ELEMENT fax-number                  (#PCDATA |
                                       (country-code?,
                                        area-code?,
                                        local-number))*>
<!ELEMENT voice-number                (#PCDATA |
                                       (country-code?,
                                        area-code?,
                                        local-number))*>
<!ELEMENT country-code                (#PCDATA)>
<!ELEMENT area-code                   (#PCDATA)>
<!ELEMENT local-number                (#PCDATA)>
<!ELEMENT sender                      (personal-name,
                                       company-name?,
                                       fax-number?,
                                       voice-number?)>
<!ELEMENT control                     (application-reference?,
                                       resolution?,
                                       priority?,
                                       coversheet?,
                                       email-notification?)>
<!ELEMENT application-reference       (#PCDATA)>
<!ELEMENT resolution                  (#PCDATA)>
<!ELEMENT priority                    EMPTY>
<!ATTLIST priority                    value
                                      (low |
                                       normal |
                                       high)
                                      "normal">
<!ELEMENT coversheet                  EMPTY>
```

```
<!ATTLIST coversheet                    value
                                        (yes |
                                         no)
                                        “yes”>
<!ELEMENT email-notification            (#PCDATA)>
<!ATTLIST email-notification            when
                                        (on-failure |
                                         always |
                                         on-success |
                                         none)
                                        "on-failure">
<!ELEMENT content                       (subject?,
                                         body+)>
<!ELEMENT subject                       (#PCDATA)>
<!ELEMENT body                          (url | bodyfile)>
<!ELEMENT url                           (#PCDATA)>
<!ELEMENT bodyfile                      (#PCDATA)>
<!ATTLIST bodyfile                      encoding-scheme
                                        (base64 |
                                         none)
                                        “base64”>
<!ATTLIST bodyfile                      mime-type
                                        (application/postscript |
                                         application/vnd.hp-PCL |
                                         image/tiff |
                                         text/plain)
                                        “text/plain”>
<!ELEMENT command-reference             (#PCDATA)>
```

The Fax-Submit-Response Document

Upon receipt of a `fax-submit` request, an XML-F service should respond by providing a `fax-submit-response`. This response is used to acknowledge receipt of the fax submission upon parsing and validation of the request. The response is used to provide a server-side reference. It contains the following structure:

```
fax-submit-response
      service-reference              (required)
      application-reference          (zero or one)
      request-status                 ("normal", warning, failed)
      request-status-code            ("Sent-Normal", Busy-pending, etc.,)
      request-status-message         (zero or one)
```

The `service-reference` is the unique reference assigned to this particular fax submission by the service provider. This is the best reference for the user application to use when requesting status or when canceling this request.

The `application-reference` is the optional (but recommended!) reference that the submitting application applied to the fax request precipitating this response. This reference may be used to get status or to cancel the request if the application has not received a `fax-submit-response` (and therefore no `service-reference` is known).

The `request-status` indicates whether the request is being processed normally, if the request is proceeding under a warning condition, or if the request has failed for some reason to be processed. The `request-status-code` provides information about the status of the fax request itself and a general purpose `request-status-message` is provided to allow an ad hoc status channel back to the application.

```
<!------------------------------------------------------------------------>
<!--XML-F Fax-Submit-Response DTD                                      -->
<!—-                                                                   -->
<!—-Request response generated upon acceptance of a fax request  -->
<!------------------------------------------------------------------------>

<!ELEMENT fax-submit-response         (service-reference,
                                       application-reference?,
                                       request-status,
                                       request-status-code,
                                       request-status-message,
                                       command-reference?)>
<!ELEMENT service-reference           (#PCDATA)>
<!ELEMENT application-reference       (#PCDATA)>
<!ELEMENT request-status              EMPTY>
<!ATTLIST request-status              value
                                      (normal |
                                       warning |
                                       failed)
                                      "normal">
<!ELEMENT request-status-code         EMPTY>
<!ATTLIST request-status-code         value
                                      ( corrupt-submission |
                                        invalid-ID |
                                        fax-accepted |
                                        fax-rejected |
                                        TBD
                                      )
                                      "normal">
<!ELEMENT request-status-message      (#PCDATA)>
<!ELEMENT command-reference           (#PCDATA)>
```

The Fax-Status Document

The `fax-status` document is used by an application to request status on a service about a particular fax request. The `fax-status` request can be used to request short and detailed status reports in either XML or text-formatted form. As well, the `fax-status` request can ask the service to mail the resulting report to an email address.

The application can either use either the `application-reference` or the `service-reference` when referring to this request. This permits the submitting application to issue status requests without having to wait or process a `fax-submit-response` document prior to asking for status. Indeed, in the event the `fax-submit-response` is never received, the application must use the application-reference to get information on the request.

The structure of the fax-status request is as follows:

```
fax-status
      account                       (required)
            id                      (required)
            subid                   (zero or one)
            mailaddress             (required)
      reference                     (required)
      format                        ("xml", formatted)
      report-type                   ("short", detail, debug)
      status-via-email              ("no", yes)
      command-reference             (zero or one)
```

```
<!-------------------------------------------------------------->
<!-- XML-F Fax-Status DTD                                     -->
<!--                                                          -->
<!—- Document used to request status on a previously submitted -->
<!—- fax request.                                              -->
<!-------------------------------------------------------------->

<!ELEMENT fax-status                  (account,
                                       reference,
                                       format,
                                       report-type,
                                       status-via-email,
                                       command-reference?)>
<!ELEMENT account                     (id,
                                       subid?,
                                       mailaddress)>
<!ELEMENT id                           (#PCDATA)>
<!ELEMENT subid                        (#PCDATA)>
<!ELEMENT mailaddress                  (#PCDATA)>
<!ELEMENT reference                     (#PCDATA)>
<!ATTLIST reference                       type
                                      (application-reference |
                                       service-reference)
                                      "service-reference">
<!ELEMENT format                      EMPTY>
<!ATTLIST format                      type
                                      (fax |
                                       formatted)
                                      #REQUIRED>
<!ELEMENT report-type                 EMPTY>
<!ATTLIST report-type                 value
                                      (short |
                                       detail |
                                       debug)
                                      "short">
<!ELEMENT status-via-email            EMPTY>
<!ATTLIST status-via-email            value
                                      (yes |
                                       no)
                                      "no">
<!ELEMENT command-reference           (#PCDATA)>
```

The Fax-Status-Response Document

The `fax-status-response` document provides the detailed information about what happened (or is happening) to a fax request on an XML-F service.

The structure for the `fax-status-response` is as follows:

```
fax-status-response
      service-reference                    (required)
      application-reference                (zero or one)
      status (one of...)                   (required)
            {short-status}
                  job-status               (new, in-process, finished)
                  job-status-disposition   (success, partial, failure)
                  job-status-message       (zero or one)
            {full-status}
                  short-status             (required)
                  attempt-status           (one or more)
                        recipient          (required)
                        date               (required)
                        called-station-id  (zero or one)
                        result-code        (required)
                        result-message     (zero or one)
            {debug-status}
                  full-status              (required)
                  fax-submit               (required)
```

```
<!-------------------------------------------------------------->
<!--XML-F Fax-Status-Response DTD                            -->
<!--                                                         -->
<!—- Used for reporting the status of a fax request         -->
<!-------------------------------------------------------------->

<!ELEMENT fax-status-response         ( service-reference,
                                         application-reference?,
                                        ( short-status |
                                         full-status |
                                         debug-status),
                                        command-reference)>
<!ELEMENT short-status                (job-status,
                                       job-status-disposition?,
                                       job-status-message?)>
<!ELEMENT job-status                  EMPTY>
<!ATTLIST job-status                  value
                                      (new |
                                       in-progress |
                                       finished)
                                      "new">
<!ELEMENT job-status-disposition      EMPTY>
<!ATTLIST job-status-disposition      value
                                      (success |
                                       partial-success |
                                       failure)
                                      "succeed">
<!ELEMENT job-status-message          (#PCDATA)>
<!ELEMENT full-status                 (short-status,
                                       attempt-status*)>
<!ELEMENT attempt-status              (recipient,
                                       date,
                                       called-station-id?,
                                       result-code,
                                       result-message?)>
<!ELEMENT recipient                    (personal-name,
                                        company-name?,
                                        fax-number,
                                        voice-number)>
<!ELEMENT personal-name                (#PCDATA)>
<!ELEMENT company-name                 (#PCDATA)>
<!ELEMENT fax-number                   (#PCDATA)>
<!ELEMENT voice-number                 (#PCDATA)>
<!ELEMENT country-code                 (#PCDATA)>
<!ELEMENT area-code                    (#PCDATA)>
<!ELEMENT local-number                 (#PCDATA)>
<!ELEMENT date                          (day, month, year,
                                         hour, minute, second)>
<!ELEMENT day                          (#PCDATA)>
<!ELEMENT month                        (#PCDATA)>
<!ELEMENT year                         (#PCDATA)>
<!ELEMENT hour                         (#PCDATA)>
<!ELEMENT minute                       (#PCDATA)>
<!ELEMENT second                       (#PCDATA)>
```

```
<!ELEMENT called-station-id              (#PCDATA)>
<!ELEMENT result-code                    EMPTY>
<!ATTLIST result-code                  value
                                       (no-answer |
                                        line-drop |
                                        busy |
                                        normal)
                                       "normal">
<!ELEMENT result-message               (#PCDATA)>
<!ELEMENT debug-status                 (short-status,
                                        full-status,
                                        fax-submit)>
```

The Fax-Cancel Document

The `fax-cancel` document is used by a sending application to cancel a previously submitted fax request. By this time the DTDs for the Fax-Cancel and the Fax-Cancel-Response should be self-describing.

```
<!-------------------------------------------------------->
<!--XML-F Fax-Cancel DTD                                -->
<!-------------------------------------------------------->

<!ELEMENT fax-cancel                  (account,
                                       reference,
                                       command-reference?)>
<!ELEMENT account                     (id,
                                       subid?,
                                       mailaddress)>
<!ELEMENT reference                    (#PCDATA)>
<!ATTLIST reference                     type
                                      (application-reference |
                                       service-reference)
                                      "service-reference">

<!ELEMENT id                          (#PCDATA)>
<!ELEMENT subid                       (#PCDATA)>
<!ELEMENT mailaddress                 (#PCDATA)>
<!ELEMENT command-reference           (#PCDATA)>
```

The Fax-Cancel-Response Document

The `fax-cancel-response` document is generated by the service upon receipt of a `fax-cancel` request. Again, the structure is similar to what has been seen:

```
<!-------------------------------------------------------------->
<!--XML-F Fax-Cancel-Response DTD                              -->
<!-------------------------------------------------------------->

<!ELEMENT fax-cancel-response         (service-reference,
                                       application-reference?,
                                       request-status,
                                       request-status-code,
                                       request-status-message,
                                       command-reference)>
<!ELEMENT service-reference           (#PCDATA)>
<!ELEMENT application-reference       (#PCDATA)>
<!ELEMENT request-status              EMPTY>
<!ATTLIST request-status              value
                                      (normal |
                                       warning |
                                       failed)
                                      "normal">
<!ELEMENT request-status-code         EMPTY>
<!ATTLIST request-status-code         value
                                      (job-terminated |
                                       too-late |
                                       cancelled-in-process |
                                       no-permission |
                                       job-not-found |
                                       TBD)
                                      "TBD">
<!ELEMENT request-status-message      (#PCDATA)>
<!ELEMENT command-reference           (#PCDATA)>
```

SAMPLE Fax Send

```
<?xml version="1.0"?>
<!DOCTYPE fax-submit SYSTEM "Fax.dtd">
<fax-submit>
      <account>
            <id>
                  Filmore5455
            </id>
            <subid>
                  David Filmore
            </subid>
            <mailaddress>
                  filemore@vsi.com
            </mailaddress>
      </account>
      <command-reference>
            xxs234234s
      </command-reference>
      <recipient>
            <personal-name>
                  Stephen D. Miller
            </personal-name>
            <company-name>
                  Willy Wonka Chocolates, Inc.
            </company-name>
            <fax-number>
                  237-0998
            </fax-number>
            <voice-number>
                  238-9873
            </voice-number>
      </recipient>
      <recipient>
            <personal-name>
                  Rob Juergens
            </personal-name>
            <fax-number>
                  <country-code>
                        011
                  </country-code>
                  <area-code>
                        987
                  </area-code>
                  <local-number>
                        234
                  </local-number>
            </fax-number>
            <voice-number>
                  234-9722
            </voice-number>
      </recipient>
      <control>
            <application-reference>
```

```
                23899
            </application-reference>
            <resolution)204x196</resolution>
            <coversheet value=”yes”/>
            <email-notification when="on-success">
                joejoe@vsi.com
            </email-notification>
        </control>
        <content>
            <subject>
                The best looking XML we are able to produce...
            </subject>
            <body>
                Some people are destined to discover that when they
                get to the end of time, there they are.
            </body>
        </content>
        <content>
            <body>
                This is some text that we think would make for a
                wonderfully interesting fax body had anyone had time
                to actually think up something fun and interesting to
                say.  Or should anyone ever decide to read it.
            </body>
        </content>
</fax-submit>
```

SAMPLE Fax Status

```
<?xml version=”1.0”?>
<!DOCTYPE fax-status SYSTEM “Status.dtd”>
<fax-status>
      <account>
            <id>
                  Filmore5455
            </id>
            <subid>
                  David Filmore
            </subid>
            <mailaddress>
                  filemore@vsi.com
            </mailaddress>
      </account>
      <command-reference>
            xxs234234s
      </command-reference>
      <reference type=”application-reference“>
            23899
      </reference>
      <format type=”fax”/>
      <report-type value=”detail”/>
      <status-via-email value=”yes”/>
</fax-status>
```

SAMPLE Fax-Cancel

```
<?xml version=”1.0”?>
<!DOCTYPE fax-cancel SYSTEM “Cancel.dtd”>
<fax-cancel>
      <account>
            <id>
                  Filmore5455
            </id>
            <subid>
                  David Filmore
            </subid>
            <mailaddress>
                  filemore@vsi.com
            </mailaddress>
      </account>
      <command-reference>
            xxs234234s
      </command-reference>
      <reference type=”application-reference“>
            23899
      </reference>
</fax-cancel>
```

For More Information See:

www.vsi.com/xml-f
www.xml-f.com

or send electronic mail to info@vsi.com

VSI (V-Systems, Inc.)
32232 Paseo Adelanto #100
San Juan Capistrano, CA 92675

(949) 489-8778
(949) 489-2486 [fax]

http://www.vsi.com