bluematter™
premium digital music

p2p Distribution Functional Specification Release 2.0

Document Number DS2.0-003

Date: March 14, 2001
Version: 0.3

Revision and Iteration History

Version	Publication Date	Authors	Summary of Changes and Updates
0.2	3/14/01	DW	Edits, 1.0/2.0 clarification
0.1	1/16/01	DW	Initial draft based on 1.0 doc and Infotects Installations instructions.

Table of Contents

1 Introduction

This document describes the bluematter 2.0 Tell-a-Friend (TAF) functionality. bluematter 2.0 supports peer-to-peer (p2p) distribution with value chain tracking.

Note: *"Tell-a-Friend" is the new MusicManager term that is synonymous with "Email-a-Friend".*

1.1 Related Documents

The following documents can be referenced for more information:

- DS2.0-006 *Email-a-Friend Server Installation*
- AR1.0-0007 *Email-a-Friend Functional Specification* (release 1.0)
- AR2.0-002 *bluematter release 2.0 Technical Specification*
- AR2.0-005 *Release 2.0 Data Flow and Definitions*
- DS*2.0*-001 *Affiliate Purchase Redirector Functional Specification*

1.2 Acronyms Used in This Document

Acronym	Meaning
BMD	bluematter data—the extension for the Package Import file
BMR	bluematter reference
CC	Content Catalog
CREF	content reference (refers to chunk)
EAF	Email-a-Friend—typically refers to the player functionality and is being replaced by Tell-a-Friend
MIME	Multipurpose Internet Mail Extension
p2p	peer-to-peer
SPOP	secure proof of purchase
TAF	Tell-a-Friend—typically refers to TAF functionality from MusicManager whereas Email-a-Friend (EAF) is a lingering term referring to this 1.0 functionality from the player/plug-in
XML	Extensible Markup Language

2 Overview

2.1 bluematter 1.0 EAF

The bluematter 1.0 release allowed a customer to send an e-mail in reference to a track from the bluematter RealJukebox skin. This e-mail enclosed a content reference (BMR file) to the offer for that track located in a database. The friend who received the e-mail double-clicked on the content reference file and was presented with an offer to purchase the track. This functionality was described in the AR1.0-0007 *Email-a-Friend Functional Specification*. EAF 1.0 functionality will still be supported under release 2.0.

2.2 bluematter 2.0 TAF Modifications

The bluematter 2.0 Tell-a-Friend functionality enhances the bluematter 1.0 functionality by tracking levels of distribution for both the affiliate and customer. Affiliate commission rates can be adjusted and customer awards programs can be implemented based on these levels of superdistribution. E-mail can be sent to a friend either through the bluematter skin or from the newly implemented MusicManager. In place of the BMR file, bluematter 2.0 uses a URL link in the body of the e-mail that links to the Affiliate Purchase Redirector.

This document focuses on p2p distribution functionality for the 2.0 release.

2.3 Overview

Figure 1 gives an overview of the Tell-a-Friend functionality.

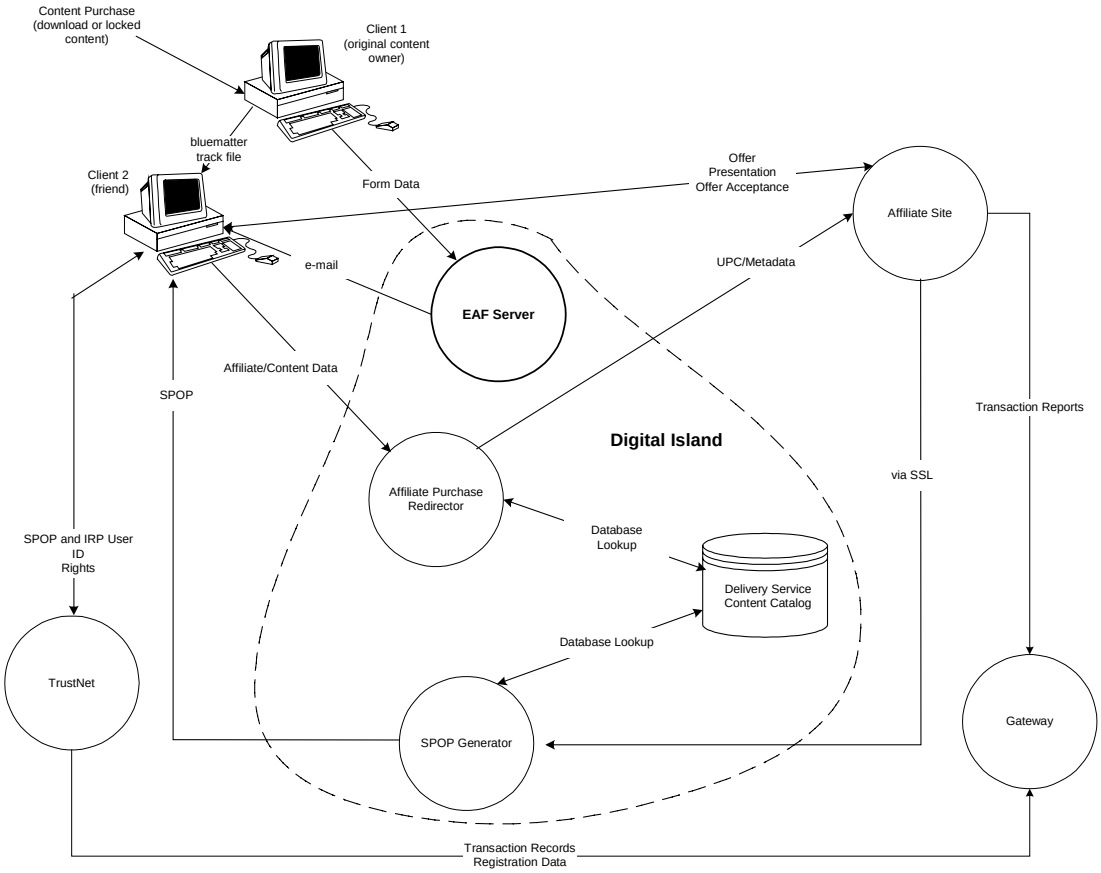

Figure 1. Tell-a-Friend Data Flow Chart

For more details on the overall system, see AR2.0-002 *bluematter release 2.0 Technical Specification.*

For EAF server installation instructions, see DS2.0-006 *Email-a-Friend Server Installation.*

1. The first customer (client 1) purchases a track. Data needed to support p2p distribution is inserted into a chunk in the track file.

2. The customer then clicks on the Tell-a-Friend button on the player skin, or initiates e-mail from MusicManager. The e-mail form is presented and sent, along with the redirect data, to the EAF server.

3. The EAF server sends an e-mail back to the friend (client 2). The e-mail presents a URL link that points to the Affiliate Purchase Redirector (APR), and includes the UPC, affiliate ID, user ID, and reference count.

4. When the recipient friend clicks on the URL link, the friend is redirected to the APR. The APR creates a link to the script and appends the UPC as part of the query string. (For more details on the APR, see DS2.0-001 *Affiliate Purchase Redirector Functional Specification*).

5. The affiliate presents a purchase page to the customer and processes the transaction, sending an XML message to the SPOP Generator.

6. The SPOP Generator creates the SPOP the same as for a normal download purchase (see DS2.0-002 *SPOP Generator Functional Specification* for more details).

7. MusicManager on the friend's computer receives the SPOP and takes the same actions as for a download purchase—getting the rights, downloading the track, and creating the chunk (see AR2.0-002 *bluematter release 2.0 Technical Specification*).

8. MusicManager loads the track into the playlist and the track plays.

3 Client Architecture

This section describes the basic functionality of the TAF client feature activated via the player/plug-in (bluematter skin) or MusicManager. The TAF client is responsible for generating the request and sending it to the EAF server.

3.1 EAF Via the Player/Plug-in

The bluematter skin for RealJukebox has a button for e-mailing a track to a friend:

Note: *The MusicMatch Jukebox and the RioPort Player have similar functionality, although not through a bluematter skin but via native functionality.*

Note: *Both 1.0 and 2.0 EAF functionality can be handled through the player/plug-in. From the client point of view there is little or no difference, but they are handled by different scripts when the request reaches the server.*

When this button is clicked, the plug-in extracts the content reference chunk from the LAFF file and dynamically builds an HTML form, embedding the content reference within it as a hidden field. The form, which appears in a browser window, contains a **send e-mail** button and other fields (see Figure 2).

Figure 2. EAF Form via the Player/Plug-in

The Email-a-Friend Web form contains the following fields:

- **From**—sender's e-mail address. This can be remembered by the plug-in.
- **To**—the recipients e-mail address
- **Subject**—brief description of e-mail contents
- **Message**—an optional message can be included
- The song's content reference—a hidden field populated by the plug-in

After filling out the form, the user clicks the **send e-mail** button. The skin does an HTTP post, leveraging the built-in Internet Explorer control.

Before calling the EAF server, the Web form does some error checking about the user input (whether it's blank or the user has a valid e-mail).

The request is processed by the EAF server (see page 11). Whether the 1.0 or 2.0 script is called depends on the version of the content.

3.2 Tell-a-Friend via the MusicManager

MusicManager contains a **Tell a Friend** button on the **Downloads** and **My Tracks** tabs. This button activates the Tell-a-Friend form (see Figure 3).

Note: *MusicManager Tell-a-Friend functionality is new to bluematter 2.0 and did not exist in 1.0.*

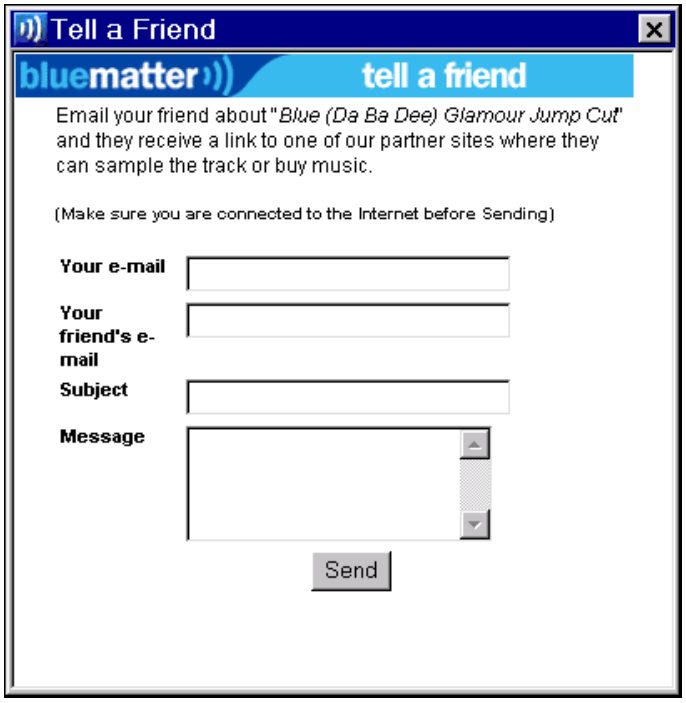

Figure 3. Tell-a-Friend Form via MusicManager

The Tell-a-Friend form was created by Digital Pulp. It has the same basic functionality as the EAF form accessed from the bluematter skin. It contains the following fields:

- **Your e-mail**—the sender's email address. This can be remembered by the plug-in.
- **Your friend's e-mail**—the recipients e-mail address
- **Subject**—brief description of e-mail contents
- **Message**—an optional message can be included
- The song's content reference—a hidden field populated by the plug-in

After filling out the form, the user clicks the **Send** button. The skin does an HTTP post, leveraging the built-in Internet Explorer control.

Before calling the EAF server, the Web form does some error checking about the user input (whether it's blank or the user has a valid e-mail).

The request is processed by the 2.0 script on the EAF server (see page 11).

3.2.1 EAF Form Sample HTML

This HTML sample is the EAF form generated from the player/plug-in. The HTML for the Tell-a-Friend form evoked by MusicManager is similar.

```
<html>
<head>
<title>E-mail &quotGee Baby, Ain't I Good To You&quot by Art Blakey to a friend!</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="javascript">
// set all cookies back to zero
function writeEmailCookie() {
  var nextyear = new Date();
  var expired = new Date();
  expired.setFullYear(expired.getYear() - 10);
  nextyear.setFullYear(nextyear.getYear() + 1);
  if (document.email_a_friend.emailcheck.checked) {
    document.cookie = "email=" + document.email_a_friend.email_from.value + "?; expires="
+ nextyear.toGMTString();
  } else if ( document.cookie.indexOf("email=") ) {
    document.cookie = "email=" + document.email_a_friend.email_from.value + "; expires="
+ expired.toGMTString();
  }
}
function readEmailCookie() {
  var allcookies = document.cookie;
  var pos = allcookies.indexOf("email=");
  resizeMe();
  if (pos != -1) {
        var start = pos + 7;
        var end = allcookies.indexOf("?", start);
        var value = allcookies.substring(start - 1, end);
        document.email_a_friend.email_from.value = value;
        document.email_a_friend.emailcheck.checked = true;
  }
}
function resizeMe() {
  window.resizeTo(450,500);
  window.focus();
}
function validate() {
  bluematter_check = "http://mail.bluematter.com/email_v100/default.asp";
  writeEmailCookie();
  var frommail = document.email_a_friend.email_from.value
  var tomail = document.email_a_friend.email_to.value
  var emailcheck = /^\w+((-\w+)|(\.\w+))*\@[A-Za-z0-9]+((\.|-)[A-Za-z0-9]+)*\.[A-Za-z0-
9]+$/;
  if ( (frommail.match(emailcheck)) == null && (tomail.match(emailcheck)) == null ) {
    alert("The TO and FROM fields contain invalid e-mail addresses.");
        //document.fixboth.submit();
  } else if ( (frommail.match(emailcheck)) == null) {
    alert ("The FROM field contains an invalid e-mail address.");
        //document.fixfrom.submit();
  } else if ( (tomail.match(emailcheck)) == null) {
    alert("The TO field contains an invalid e-mail address.");
    //document.fixto.submit();
```

```
  } else if (bluematter_check == "null") {
    alert("Sorry, you may only email bluematter songs");
    //document.sorryForm.submit();
  } else {
    document.email_a_friend.submit();
  }
}
</script>
</head>
<body bgcolor="#FFFFFF" background="background.gif" topmargin="0" leftmargin="0"
onload="readEmailCookie()">
<form name = "sorryForm" method="post" action ="file://d:\Program
Files\Real\RealJukebox\temp\~rjbtemp20\template8.html"></form>
<form name = "fixfrom" method="post" action =""></form>
<form name = "fixto" method="post" action =""></form>
<form name = "fixboth" method="post" action =""></form>
<form name="email_a_friend" method="post" action="http://universal-sun-7.digisle.net/p2p-
bin/P2P.pm" target="new">
  <table border="0" cellspacing="1" cellpadding="0" align="center" width=380>
  <tr>
        <td width="75"></td>
        <td width="230" height="50"><div align="center"><font face="Trebuchet
MS,Verdana,Arial,Helvetica"><b>E-mail &quotGee Baby, Ain't I Good To You&quot by Art
Blakey to a friend!</b></font><br><font face="Trebuchet MS,Verdana,Arial,Helvetica"
size="2"><center>You must be online to use this feature</center></font></div></td>
        <td width="75" rowspan=9></td>
  </tr>
  <tr>
        <td width="75"></td>
        <td width="230"><font face="Trebuchet MS,Verdana,Arial,Helvetica"
size="1"><b>Your E-mail address:</b></font></td>
</tr>
<tr>
        <td width="75"><div align="right"><font face="Trebuchet
MS,Verdana,Arial,Helvetica" size="2" color="#7f080b"><b>FROM:</b></font></div></td>
        <td width="230"><input name="email_from" type="text" size="30"></td>
</tr>
<tr>
        <td width="75"></td>
        <td width="230"><div align="center"><input type="checkbox" name="emailcheck"
value="emailcheck" checked><font face="Trebuchet MS,Verdana,Arial,Helvetica" size="1"
color="#7f080b"><b> Remember my e-mail address.</b></font></div></td>
</tr>
<tr>
        <td width=380 height=5 colspan=3></td>
</tr>
  <tr>
        <td width="75"></td>
        <td width="230"><font face="Trebuchet MS,Verdana,Arial,Helvetica"
size="1"><b>Recipient's E-mail address:</b></font></td>
</tr>
<tr>
        <td width="75"><div align="right"><font face="Trebuchet
MS,Verdana,Arial,Helvetica" size="2" color="#7f080b"><b>TO:</b></font></div></td>
        <td width="230"><input name="email_to" type="text" size="30"></td>
</tr>
<tr>
        <td width="75"><div align="right"><font face="Trebuchet
MS,Verdana,Arial,Helvetica" size="2" color="#7f080b"><b>SUBJECT:</b></font></div></td>
        <td width="230"><input name="message_subject" type="text" size="35"
maxlength="100"></td>
<tr>
        <td width="75"><div align="right"><font face="Trebuchet
MS,Verdana,Arial,Helvetica" size="2" color="#7f080b"><b>MESSAGE:</b></font></div></td>
        <td width="230"><textarea name="message_body" rows="4" cols="30"></textarea>
</tr>
<tr>
        <td width="75"></td>
          <td width="230"><br><div align="center">
```

```
       <a href="javascript:validate()"> <IMG SRC="send.gif" border="0" width=172
height=23></a></td>
</tr>
       <input name="title" type="hidden" value = "Gee Baby, Ain't I Good To You">
       <input name="artist" type="hidden" value = "Art Blakey">
       <input name="upc" type="hidden" value = "044001015021">
       <input name="reference_count" type="hidden" value = "1">
       <input name="user_id" type="hidden" value="123">
       <input name="affiliate_id" type="hidden" value="2">
       </div>
</table>
</form>
</body>
</html>
```

3.3 Data Extracted by the EAF Client

Besides the data gathered from the form, the MusicManager or player plug-in extracts the data listed in Table 1. This data comes from the Chunk_Info chunk.

Table 1. Data Extracted from Chunk_Info

Data Field	Definition
UPC	Uniform Product Code or Product ID
AFFILIATE_ID	The identifier for the e-tailer or channel
USER_ID	MusicManager inserts the IRP user ID as the referring member ID to support future p2p distribution. If it is blank, it is ignored.
REFERENCE_COUNT	p2p level. For the initial purchase, MusicManager inserts a value of 1 for the reference count. For subsequent p2p purchases, MusicManager increments the value of the existing counter by 1.

Error checking is performed on user input (no blanks and valid e-mail).

3.4 Direct File Transfer

Rather than use the TAF or EAF feature, the first client may send the BMT directly to the friend. This method is summarized here for completeness.

1. A customer purchases a bluematter track and gives the BMT file to a friend, via physical means such as a Zip disk or an e-mail attachment.

2. The recipient clicks on the track. If the person has a bluematter-compatible player the track is imported into the playlist.

3. When the customer attempts to play the track, the secure portion of the plug-in (`snitrn.dll`) sees that rights to play the track are not in the protected database. It displays a message stating that the customer does not have the rights to the track and provides a button or a link for the customer to purchase the track.

4. If the customer opts to buy the track, the secure portion of the plug-in calls MusicManager and passes the path to the file as an argument.

5. MusicManager opens the file, imports the data into the track data file, and connects to the APR.

6. The APR creates a link to the script and appends the UPC as part of the query string. (For more details on the APR, see DS*2.0*-001 *Affiliate Purchase Redirector Functional Specification*).

7. The affiliate presents a purchase page to the customer and processes the transaction, sending an XML message to the SPOP Generator.

8. The SPOP Generator creates the SPOP the same as for a normal download purchase (see DS2.0-002 *SPOP Generator Functional Specification* for more details).

9. MusicManager on the friend's computer receives the SPOP and takes the same actions as for a download purchase, getting the rights, downloading the track, and creating the Chunk_Info chunk (see AR2.0-002 *bluematter release 2.0 Technical Specification*).

10. MusicManager determines that the track is already on the hard drive, loads it into the player's playlist, and plays the track.

11. The track begins to play.

3.5 bluematter Version Detection

The player plug-in (`sncrrn.dll`) determines the format and version of the content. The following logic is used.

1. The player plug-in assumes the content is bluematter 2.0 and attempts to extract data from Chunk_Info.

2. If the player plug-in does not detect the Chunk_Info chunk, then it looks for the content reference (CREF) chunk. If it finds the CREF chunk then it assumes it is bluematter 1.0 and creates a BMR.

3. If neither the Chunk_Info or CREF chunks are found, then the player plug-in assumes it is locked content.

3.6 Registry Settings

On order to support 2.0 tracks, MusicManager sets up the registry key

```
HKEY_CLASSES_ROOT\Software\Bluematter\1.0\Preferences\EmailUrl
```

if it already wasn't created during the installation.

For 1.0 tracks, the registry setting is:

```
HKEY_CLASSES_ROT\Software\RealNetworks\RealJukebox\1.0\Preferences\
BlueMatter\email_action
```

3.7 Non-bluematter Files

Non-bluematter files, that do not have the email_friend asset, are not supported by bluematter. If a non-bluematter track is playing when the Email-a-Friend button is clicked, a browser window opens displaying the following message:

```
“e-mail a friend” is not yet supported for this track.
```

The player plug-in detects the presence of the Chunk_Info chunk to determine whether EAF is supported.

3.8 Other Incompatibility Issues

If a BMT is directly sent to a person who does not have the required bluematter plug-in, and the customer attempts to access this file, the player plug-in displays an error message with a reference to contact the bluematter Web site. This is a Customer Care issue.

If a 2.0 style TAF is sent to a 1.0 user, then the 1.0 user could purchase the track. When the 1.0 user tries to play the track, the player plug-in would direct the user to obtain a new 2.0 version of the plug-in.

4 Server Architecture

For the 1.0 release there was one live EAF server running on NT (written as an active server page (ASP)). For 2.0, this architecture is replaced by a UNIX server running a script that still supports EAF 1.0, and a separate script to support TAF 2.0.

4.1 Release 2.0 EAF Server

The 2.0 release includes two different EAF scripts running on the same UNIX server (both written in mod_perl):

- One EAF script handles the old bluematter 1.0 EAF functionality (via the RealJukebox skin). See page 13.
- Another EAF script handles the bluematter 2.0 TAF functionality (via the MusicManager button or the player skin)

The 2.0 release is a drop-in replacement that is Solaris-based and is intended to be run on a SPARC-compatible server residing at Digital Island. Apache is used as the Web server. Its address is a domain name so that if the server moves, or a different server is selected, its just a matter of

updating the domain name information to point to the new IP address. The EAF server is designed around a two-box solution that is load-balanced and redundant.

The EAF application is written as a Perl module and uses the Apache mod_perl embedded interpreter for greatest efficiency. The mail server used is Sendmail.

4.2 2.0 TAF Script Implementation

The EAF server inputs the data in Table 1 on page 9 and exports the data back to the second customer (the friend) in the form of an e-mail with a URL reference link to the track.

1. When the EAF server receives a request for the TAF service, Apache passes this request to the Perl application.

2. The `P2P.pm` module parses the request and verifies that input parameters are correct. Any user input errors result in an appropriate message returned to the user. Any other internal errors are logged in an error log and an internal error message is returned to the user (see the *Logging Transactions* section on page 14).

3. The script writes a record to a SQL database containing the following:

- Time
- **From** person's e-mail address
- **To** person's e-mail address
- Subject
    - Message
    - Data from the content reference such as the ID

4. The script generates the e-mail and sends it to the friend.

5. A URL link is created to the Affiliate Purchase Redirector (APR) within the e-mail. The Affiliate ID, UPC, User ID, and Reference Count are embedded into this URL.

6. SMTP is used to send the message to the Sendmail server. The SMTP format is flexible, making it easy to switch e-mail servers if desired. The e-mail is sent to the friend.

   A sample of this 2.0 style e-mail is shown in Figure 4.

Figure 4. Sample e-mail Sent to Friend with 2.0 Style Content Reference URL

7. Upon successful completion, the basic request parameters are logged in the database and a **Success** message is returned to the first customer.

8. When the friend clicks on the reference URL, the Affiliate Purchase Redirector processes the transaction and redirects the customer to the affiliate shopping cart to purchase the track. See DS2.0-001 *Affiliate Purchase Redirector Functional Specification* for more details on the rest of the process.

4.3 Release 1.0 EAF Script

The 2.0 release will continue to support the 1.0 style EAF as follows:

1. The user input is obtained via the client (the bluematter skin, see page 4 for more details). The HTML form contents are sent to the EAF server.

2. The EAF 1.0 Script parses the contents and extracts the offer ID.

3. An e-mail is generated with a BMR attached. This e-mail is sent to the second client (the friend).

Figure 5. Release 1.0 e-mail with BMR Attachment

4. Any errors are logged to a file. In addition, each transaction is logged to the Oracle database. See the *EAF 1.0 Transaction Logs* section on page 14.

5. An HTML page is returned to the first client showing the status of the send operation, indicating success or failure, and in the case of failure, the reason for the failure.

6. When the e-mail recipient or friend clicks the BMR file attachment, RealJukebox opens and an offer is downloaded from the Reference Service.

Sample contents of the BMR content reference chunk for 1.0 are shown in Figure 6.

```
<content-reference version="1.0.0">
  <item>
    <ID DRID="UMG" CHID="00008" OFID="0000800000005180"
       PRID="044001015021" OBID="USMC16147539" PLID=""/>
    <DESC title="Gee Baby, Ain't I Good To You" artist="Art Blakey"/>
  </item>
</content-reference>
```

Figure 6. Content Reference Chunk Sample for 1.0

5 Logging Transactions

The EAF server logs transactions into separate files for 1.0 and 2.0.

5.1 TAF 2.0 Transaction Log

2.0 transactions are logged into the `P2Ptmp.log` file. This is a flat text file that logs all TAF 2.0 actions regardless of whether they passed of failed. The following fields are recorded in this log:

- Timestamp
- Email_From
- Email_To
- Affiliate_Id
- UPC

5.2 EAF 1.0 Transaction Logs

Successful 1.0 attempts are logged in the Oracle database table EAF_LOG.

Unsuccessful attempts are logged in the `EAFtmp.log` file and describe why the request was unsuccessful. This includes all kinds of errors like Oracle database failure, failure to mail, XML parser error, etc. It is the responsibility of the system administrator to manage this log (archival, size limiting, etc.).

The following parameters are logged (see the `eaf_log.sql` file for exact table definition):

- RQSTID—this is the internal request id generated at the time of insert
- LOGTIME—this is the date the request was logged
- CONTREFID—this is the content reference ID (song ID) referred
- EMAIL_TO—this is who the referral was sent to
- EMAIL_FROM—this is who sent the referral
- FORMID—this is the skin ID used in the request