gSPECIFICATION	REV.
484-0200155	F5

NCR Corporation
Image & Payment Systems
50 Northland Road
Unit 100
Waterloo, Ontario
N2V1 N3

PROGRAM: ImageMark Archive 5.1

TITLE: **Interface Specification – Archive Content Services**

DATE: December 7, 2016

RELEASE: Draft

SOURCE ORGANIZATION: Solutions Architecture

Prepared By:

Peter Robinson, Solutions Architecture Date

Approved By:

Judy Sandison, Manager, Solutions Architecture Date

SPECIFICATION	REV.
484-0200155	F5

NCR Corporation
Image & Payment Systems
50 Northland Road
Unit 100
Waterloo, Ontario
N2V1 N3

PROGRAM: ImageMark Archive 5.1

TITLE: **Interface Specification – Archive Content Services**

DATE: December 7, 2016

RELEASE: Draft

SOURCE ORGANIZATION: Solutions Architecture

CHANGE SHEET

Rev	Date	Section	Description of Change	By
A	06/13/2003	All	Initial Release – 53DR25561	Peter Robinson
B	09/03/2003	All	Change Release – 53DR25690	Peter Robinson
C	11/30/2003	All	Change Release – 53DR25905	Peter Robinson
D	10/28/2004	All	Change Release – 53DR26456	Peter Robinson
E	02/16/2005	All	Change Release – 53DR24994	
F1	06/01/2005			Peter Robinson
F		All	Change Rev.F is a copy of draft Rev.Fn	Peter Robinson
F		All	Released on 53DRnnnnn	Peter Robinson
F2	03/28/2006	6.2	Added error codes for Fill element	
F3	09/22/2014	All	Updated 5.1 Release Information	Saurabh Patel
F4	10/17/2016	6, 7, 12	New sections	Anjali Phatak
F5	11/18/2016	13	New section - FAQs	Anjali Phatak

TABLE OF CONTENTS

1. Summary

This specification defines the interface between ImageMark Archive Content Services (ACS) and NCR or third-party applications and services that need to use ACS to access ImageMark Archive.

Although oriented toward the retrieval and processing of object content, the ACS interface provides access to any of the ImageMark Archive services that are currently available through Archive Interaction Services (AIS). In the case of AIS services that return object content, the requester may ask for this content to be processed in specified ways before being returned. This processing may include selecting parts of the content, transcoding from one format to another and performing image manipulation functions on raster scan content.

ACS provides an HTTP interface that lets an application or service send an HTTP POST message request to ACS. This message contains an XML body that defines the service required. As well as containing regular AIS request elements, this XML may contain ACS-specific elements that define how content data is to be processed before being returned to the requester. ACS passes this through to AIS, retrieves the result, performs any requested processing, then returns the result to the requester as an HTTP response.

1.1 Nomenclature and Notes on Reading this Document

Within this interface specification, the following nomenclature is used:

- The word "SHALL" is used to indicate a binding function that must be performed by the system implementing the interface. The word "will" is used to describe additional details of the binding function.
- In the leftmost column of this document, each function has a unique identifier in the form: Rxx. The 'R' refers to the revision level of the document in which the function was first introduced or last changed. The 'xx' is a sequence number, starting at 1 within each section. As an example, the first function in each section of the document's initial release will be "A1", the second function will be "A2", etc.
- When an existing function is changed between revisions, only the revision level changes; the sequence number remains the same. When a new function is added, it is assigned the new revision level and the next new sequence number in the section. If a function is deleted within a section, its sequence number can no longer be used in that section. The change log will reflect all functions changed, added or deleted.

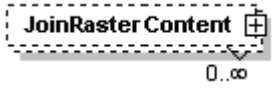
This document uses the XML Schema notation to define the format of XML requests and responses. It is assumed that the reader is familiar with this. If not, reference [4], *XML Schema Part 0: Primer* provides a useful introduction.

Some of the figures in this specification were automatically generated from XML Schemas using the **XMLSpy Schema Editor**. The conventions used in these figures are shown below:

A required XML element that must appear once.

An optional XML element that may appear once.

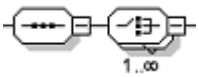
A required XML element that must appear one or more times. The maximum number of times that an item can occur is shown by a range `1..n`, where `1..∞` indicates no limit.



An optional XML element that may appear zero or more times. The maximum number of times that an item can occur is shown by a range `0..n`, where `0..∞` indicates no limit.

A sequence. The elements to the right of this symbol appear in the order shown.

A choice. Only one of the elements to the right of this symbol may appear.



By using the sequence followed by a multiple choice symbol, we can show a sequence of choices. This denotes a sequence of elements, each of which may be one of the possible choices.

The ⊞ in an element symbol indicates that symbol is a collapsed tree that will be shown in an expanded form in another figure. The little arrow ↗ can be ignored. It means that, within the XML document, the element is defined as a reference to another element and not explicitly included as a separate element.

2. References

2.1 Referenced Documents

1. “ImageMark Archive Retrieval – Interface Specification – Retrieval Interface”, 484-0028629, Rev.F, NCR I&PS, December 16 ,2003. Available at http://www.info.ncr.com
2. "ImageMark Capture: Capture System to ImageMark Archive Interface Specification", 484-0200265, Rev.B, NCR I&PS, February 8, 2005. Available at http://www.info.ncr.com
3. “Extensible Markup Language (XML) 1.0 (Third Edition)”, World Wide Web Consortium, February 4, 2004. Available at http://www.w3.org/TR/REC-xml/
4. “XML Schema Part 0: Primer Second Edition”, World Wide Web Consortium, October 28, 2004. Available at http://www.w3.org/TR/xmlschema-0/
5. “XML Schema Part 1: Structures Second Edition”, World Wide Web Consortium, October 28, 2004. Available at http://www.w3.org/TR/xmlschema-1/
6. “XML Schema Part 2: Datatypes Second Edition”, World Wide Web Consortium, October 28, 2004. Available at http://www.w3.org/TR/xmlschema-2/
7. “Multipurpose Internet Mail Extensions(MIME) Part One: Format of Internet Message Bodies”, RFC 2045, Internet Engineering Task Force (IETF), November 1996. Available at http://www.ietf.org/rfc/rfc2045.txt
8. "Hypertext Transfer Protocol – HTTP/1.1", RFC 2616, Internet Engineering Task Force (IETF), June 1999. Available at http://www.ietf.org/rfc/rfc2616.txt
9. "Hypertext Transfer Protocol – HTTP/1.0", RFC 1945, Internet Engineering Task Force (IETF), May 1996. Available at http://www.ietf.org/rfc/rfc1945.txt
10. "X9.100-140-2004: Specifications for an Image Replacement Document – IRD", Accredited Standards Committee X9, October 1, 2004.
11. "TIFF Revision 6.0 Specification", Adobe Systems Incorporated, June 3, 1992. Available at http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf
12. "TIFF Technical Note #2 (JPEG in TIFF)", March 17, 1995, Available at ftp://ftp.sgi.com/graphics/tiff/TTN2.draft.txt.
13. "Mixed Object Document Content Architecture (MO:DCA) Reference ". SC31-6802-05, International Business Machines Corporation, 6th Edition – April 2001, Available at http://publibz.boulder.ibm.com/epubs/pdf/c3168025.pdf

3. Archive Content Services Interfaces

Archive Content Services lets a client program request ImageMark Archive services through an HTTP Interface.

3.1 Conventions

All examples of XML in this document are shown in `courier` font.

3.2 HTTP Interface

C5 All communication between client programs and Archive Content Services SHALL use the HTTP/1.0 or HTTP/1.1 protocols as defined in reference [9], *Hypertext Transfer Protocol – HTTP/1.0* and reference [8], *Hypertext Transfer Protocol – HTTP/1.1*.

C1 A client program SHALL send a request to Archive Content Services as a Content-Type `text/xml` body of an HTTP POST request.

A2 The ACS request SHALL be an XML document in the format defined in section 3.3, *Archive Request Format*.

A3 ACS SHALL return the response to the requester as a Content-Type `text/xml` body of an HTTP response.

E4 The ACS response SHALL be an XML document in the format defined in section 3.4, Archive Result Format

Example:

A typical POST method request to retrieve a check item, extract an image from the item, then rotate this image through 90° might be:

```
POST /acs/servlet/acs HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows NT)
Host: www.foo.ncr.com
Accept: text/xml,text/html
Content-Type: text/xml
Content-Length: 2134

<?xml version="1.0" encoding="UTF-8"?>
<inquiry sectoken="4db3afda35…" applname="app1" sessionname="session1" images="Y">
    <object name="Commercial"/>
    <resultfield name="postingdate"/>
    <resultfield name="account_number"/>
    <sortfield name="account_number" order="ASC"/>
    <indexquery queryname="test query 1">
        <qfield name="postingdate" low="20000101" high="20031231"/>
        <qfield name="account_number" low="0" high="999999999"/>
        <ProcessContent>
            <GetImagePage Pages="1">
                <ManipulateRasterContent>
                    <Rotate Degree="90"/>
                </ManipulateRasterContent>
            </GetImagePage>
                …
        </ProcessContent>
    </indexquery>
</inquiry>
```

3.3 Archive Request Format

3.3.1 General

C1 A client program SHALL be able to request any service that is provided by Archive Interaction Services (AIS) by submitting an XML request document in the format and sequence defined in reference [1], *ImageMark Archive Retrieval – Interface Specification – Retrieval Interface* to Archive Content Services.

E2 If the request is for one of the AIS services that can return object content, a client program SHALL be able to ask ACS to process this content in defined ways before returning it. This is to be done by adding ACS-specific `<ProcessContent>` elements to the AIS request. For any requested content processing to be meaningful, the request must actually result in content being returned.

In the context of this version of the specification, "content" will restricted to NCR TIFF and MO:DCA content associated with a retrieved object. However, this restriction will not apply in future and content may include non-image types of data.

The following AIS services can return object content:

- **Inquiry Service** – In the `indexquery` function.
- **Batch Access Service** – In the `getbatch` and `getimage` functions.

Note:

If an Inquiry Service batch request has been instructed to send result files to a specified host instead of making them available through the Batch Access Service, Archive Content Services cannot be used to process the content resulting from this request.

A3 In all cases, the `<ProcessContent>` element SHALL have the format defined in section 3.3.4, *Element <ProcessContent> Format*.

C4 All elements in XML request documents SHALL be well-formed XML as defined in reference [3] *Extensible Markup Language (XML) 1.0 (Third Edition)*

C5 *Deleted.*

3.3.2 Requesting Content Processing – AIS Inquiry Service

A1 A client program SHALL be able to specify that ACS process retrieved object content by adding a `<ProcessContent>` element to an AIS Inquiry Service request.

C2 The client program SHALL be able to add the `<ProcessContent>` element in the following places in the Inquiry Service request.

- To the `<inquiry>` element
- To one or more of the `<indexquery>` child elements within the `<inquiry>` element

C3 If the `<ProcessContent>` element is added to the `<inquiry>` element, preceding or following all the `<indexquery>` element, as shown in the XML example below, the processing it defines SHALL be applied to the content returned by **all** of the `<indexquery>` child elements in the Inquiry Service Request.

```
<?xml version="1.0" encoding="UTF-8"?>
<inquiry sectoken="4db3afda35…" applname="app1" sessionname="session1" images="Y">
    <object name="Commercial"/>
    <resultfield name="postingdate"/>
    <resultfield name="account_number"/>
    <sortfield name="account number" order="ASC"/>
    <ProcessContent>
        content processing instructions
    </ProcessContent>
    <indexquery queryname="test query 1">
        <qfield name="postingdate" low="20000101" high="20031231"/>
        <qfield name="account_number" low="000" high="554"/>
    </indexquery>
    <indexquery queryname="test query 2">
        <qfield name="postingdate" low="20000101" high="20031231"/>
        <qfield name="account_number" low="555" high="999"/>
    </indexquery>
</inquiry>
```

A4 If the <ProcessContent> element is added to an <indexquery> element as shown in the XML example below, the processing it defines SHALL be applied to the content returned by that <indexquery> element only.

```
<?xml version="1.0" encoding="UTF-8"?>
<inquiry sectoken="4db3afda35…" applname="app1" sessionname="session1" images="Y">
    <object name="Commercial"/>
    <resultfield name="postingdate"/>
    <resultfield name="account_number"/>
    <sortfield name="account_number" order="ASC"/>
    <indexquery queryname="test query 1">
        <qfield name="postingdate" low="20000101" high="20031231"/>
        <qfield name="account_number" low="0" high="999999999"/>
        <ProcessContent>
            content processing instructions
        </ProcessContent>
    </indexquery>
</inquiry>
```

B5 If a <ProcessContent> element is added both to the <inquiry> element and its <indexquery> child element in an AIS Inquiry Service Request, the processing defined for the <indexquery> child element SHALL take precedence over, and be used instead of, the <ProcessContent> defined for the higher-level <inquiry> element.

A6 If an <indexquery> element returns content for more than one object, the processing defined in the <ProcessContent> element SHALL be applied to the content for every object. The objects are to be output in the same order that they would have been output had processing not been done.

C7 If more than one <ProcessContent> element appears in an <inquiry> element or an <indexquery> element, the first <ProcessContent> element shall be the one that is acted on. Subsequent <ProcessContent> elements in the same <inquiry> or <indexquery> element SHALL be logged to the error log but otherwise ignored.

3.3.3 Requesting Content Processing – AIS Batch Access Service

A1 A client program SHALL be able to specify that ACS process retrieved object content by adding a <ProcessContent> element to an AIS Batch Access Service request.

C7 The client program SHALL be able to add the <ProcessContent> element in the following places in the Batch Access Service request.

- To the <batchaccess> element
- To one or more of the <getbatch> child elements within the <batchaccess> element
- To one or more of the <getimage> child elements within the <batchaccess> element

B8 If the <ProcessContent> element is added to the <batchaccess> element as shown in the XML example below, the processing it defines SHALL be applied to the content of all objects returned by all of the <getbatch> or <getimage> child elements in the Batch Access Service Request.

```
<?xml version="1.0" encoding="UTF-8"?>
<batchaccess sectoken="4db3afda35…" applname="app1" sessionname="session1">
    <ProcessContent>
        content processing instructions
    </ProcessContent>
    <getbatch batchname="test_batch_1" images="Y">
        <qfield name="acct_no" low="12345" high="54321"/>
    </getbatch>
    <getbatch batchname="test_batch_2" images="Y">
        <qfield name="acct_no" low="12345" high="54321"/>
    </getbatch>
</batchaccess>
```

B2 If the <ProcessContent> element is added to a <getbatch> child element of a <batchaccess> element as shown in the XML example below, the processing it defines SHALL be applied to the content of all objects returned by that <getbatch> element only.

```
<?xml version="1.0" encoding="UTF-8"?>
<batchaccess sectoken="4db3afda35…" applname="app1" sessionname="session1">
    <getbatch batchname="test_batch_1" images="Y">
        <qfield name="acct_no" low="12345" high="54321"/>
        <ProcessContent>
            content processing instructions
        </ProcessContent>
    </getbatch>
</batchaccess>
```

B4 If the <ProcessContent> element is added to a <getimage> child element of a <batchaccess> element as shown in the XML example below, the processing it defines SHALL be applied to the content of all objects returned by all the <imagehandle> child elements in that <getimage> element only.

```
<?xml version="1.0" encoding="UTF-8"?>
<batchaccess sectoken="4db3afda35…" applname="app1" sessionname="session1">
    <getimage batchname="test_batch_1">
        <ProcessContent>
            content processing instructions
        </ProcessContent>
        <imagehandle id=”685a8h3y…”/>
        <imagehandle id=”686g3h3y…”/>
        <imagehandle id=”687b3h3y…”/>
    </getimage>
</batchaccess>
```

C10 *Deleted.*

C9 If a <ProcessContent> element is added to both a <batchaccess> element and to one or more of its <getbatch> or <getimage> child elements in an AIS Batch Access Service Request, the processing defined for the <getbatch> or <getimage> child element SHALL take precedence over, and be used instead of, the <ProcessContent> defined for the higher-level <batchaccess> element.

C11 *Deleted*

C3 *Deleted.*

C12 If more than one <ProcessContent> element appears in a <batchaccess>, <getbatch> or <getimage> element, the first <ProcessContent> element SHALL be the one that is acted on. Subsequent <ProcessContent> elements in the same <batchaccess>, <getbatch> or <getimage> element SHALL be logged to the error log but otherwise ignored.

3.3.4 Element <ProcessContent> Format

For this release of Archive Content Services, manipulation and transcoding of object content will only be available for NCR TIFF and MO:DCA objects through the `<GetImagePage>` element. Section 3.3.4 has been left in the specification because it is anticipated that the more general ability to manipulate and transcode other types of content will implemented in later releases.

A1 The `<ProcessContent>` element SHALL define the processing to be performed on object content returned by Archive Interaction services.

A2 The `<ProcessContent>` element SHALL provide content manipulation functions on raster scan content as defined by the `<ManipulateRasterContent>` child element.

A3 The `<ProcessContent>` element SHALL allow the transcoding of object content from one format to another as defined by the `<TranscodeContent>` child element.

E4 The `<ProcessContent>` element SHALL be able to extract specified pages from a multi-page TIFF or MO:DCA object as defined by the `<GetImagePage>` child element in section 3.3.6, Element <GetImagePage> Format.

B5 *Deleted.*

C6 *Deleted.*

A7 The child elements of the `<ProcessContent>` element that define the content processing SHALL be processed in the order that they appear. It is the responsibility of a client program to order them so that desired content processing is achieved. Any processing element may appear more than once if this is necessary to perform the required process.

E8 The format of the `<ProcessContent>` element SHALL be as defined in the XML Schema that is documented in section 4, *Appendix A: Element <ProcessContent> XML Schema*.

An example of a `<ProcessContent>` element is shown below. This example defines processing to be applied to the content of a single retrieved object. Figure 3.1 shows the structure of this element diagrammatically.

```
<ProcessContent>
    <ManipulateRasterContent>
        <Clip Unit="inches"
              HeightOrigin="top"
              HeightStart="0.0"
              Height="2.27"
              WidthOrigin="left"
              WidthStart="2.0"
              Width="2.33"/>
        <Rotate Degree="90"/>
    </ManipulateRasterContent>
    <TranscodeContent RequestedContentDescriptor="PNG"/>
</ProcessContent>
```

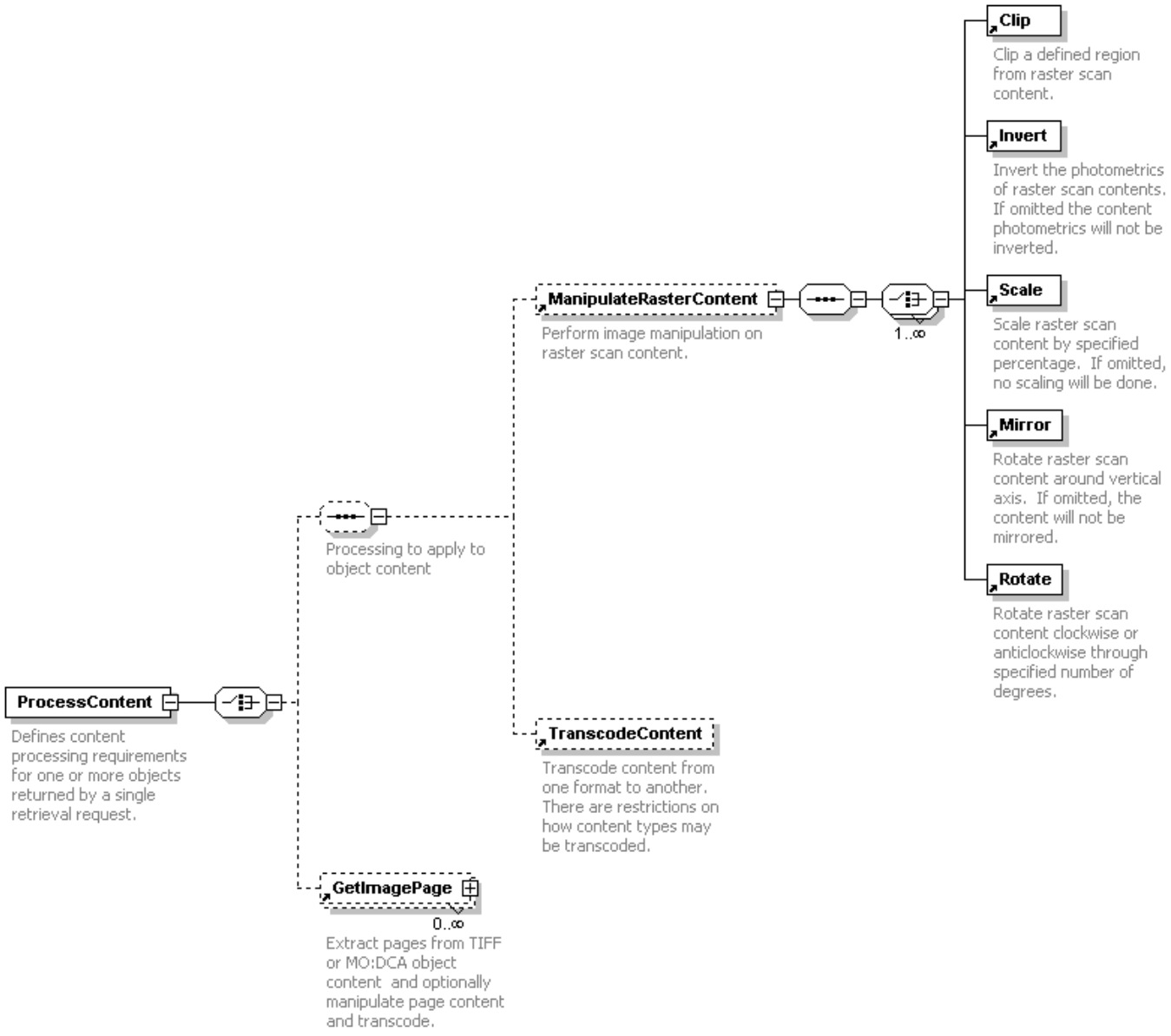


Figure 3-1: Element `<ProcessContent>` Structure

3.3.5 Element <GetTiffPage> Format

E1 Element `<GetTiffPage>` shall be replaced by element `<GetImagePage>` which is to have exactly the same format and is to provide exactly the same functionality, as defined in section 3.3.6, Element <GetImagePage> Format , but is to extend it to processing MO:DCA object content.

E2 For reasons of back-compatibility, element `<GetTiffPage>` shall still be accepted as valid input but is to be treated in every way as a synonym for `<GetImagePage>`.

E3 If element `<GetTiffPage>`, instead of `<GetImagePage>` is used in a request, for reasons of back-compatibility the result element shall be `<GetTiffPageResult>` not `<GetImagePageResult>`.

Note: *All other requirements previously defined in this section have now been moved to section* 3.3.6, Element <GetImagePage> Format.

3.3.6 Element <GetImagePage> Format

For this release of Archive Content Services, manipulation and transcoding of object content will only be available for NCR TIFF and MO:DCA objects through the `<GetImagePage>` element.

E1 The `<GetImagePage>` element SHALL allow one or more pages to be extracted from a multi-page TIFF or MO:DCA object.

E2 The `<GetImagePage>` element SHALL provide optional content manipulation functions by performing image manipulation on the raster scan content of TIFF or MO:DCA pages as defined by the `<ManipulateRasterContent>` child element.

E3 The `<GetImagePage>` element SHALL allow optional transcoding of object content from one format to another as defined by the `<TranscodeContent>` child element.

E4 If any content manipulation functions are requested, the content shall be automatically transcoded to PNG unless requested otherwise by a `<TranscodeContent>` element.

E5 The format of the `<GetImagePage>` element SHALL be as defined in the XML Schema file `ProcessContent.xsd`.

E6 The page or pages to be extracted from the TIFF or MO:DCA object SHALL be defined by an optional `Pages` attribute, where the first page in the file is page 1. The value of this attribute may be:

- A **specific page** number, e.g. `Pages="3"` or
- A **page range** which may be a **closed range** or an **open range**, e.g. `Pages="5-11"` for a closed range or `"3-"` for an open range or
- A comma-separated list of pages/page ranges, e.g. `Pages="3,6,8-15,17"`

E7 If the `Pages` attribute is omitted, all pages in the object SHALL be extracted.

E8 A null value for the `Pages` attribute (`Pages=""`) SHALL be an error.

E9 Page ranges SHALL be specified by a `Pages` attribute value in the format `Pages="`*`low`*`-`*`high`*`"`, where *`low`* is the first page to be retrieved and *`high`* is the last page.

E10 In a page range, the value for *`low`* SHALL NOT be greater than the value for *`high`*. E.g. `Pages="4-2"` is an error.

E11 If the *`high`* value is omitted from a page range, the range is an open range and all pages in the object starting with the *`low`* value SHALL be returned. E.g. In an object with 12 pages, the attribute `Pages="3-"` will return pages 3 through 12.

E12 If a closed page range includes pages that do not exist in the object, those pages that do exist SHALL be retrieved.. E.g. In an object with only 12 pages, the attribute `Pages="3-99"` will retrieve pages 3 through 12.

E13 In a list of pages/page ranges, the pages SHALL be retrieved in the order they appear in the list.

E14 There SHALL be no restriction in the order that pages appear in a list. Duplicates or overlapping ranges are to be allowed and will result in more than one instance of the content for a page being returned. E.g. the attribute `Pages="9,3,2-6,5,5,3-7,1"` is valid.

E15 Page numbers SHALL be pages that exist in the object. A request for a non-existent page is an error.

E16 If a **closed range** includes pages that do not exist in the object, this SHALL be an error. E.g. In an object with only 12 pages, the attribute `Pages="3-99"` is an error. Pages 3 through 12 will still be retrieved, but an error will be logged in the log file to indicate the absence of pages 13 through 99.

E17 If an **open range** has a *`low`* value for the `Pages` attribute that is higher than the highest page in the object, this SHALL be an error. E.g. In an object with only 12 pages, the attribute `Pages="15-"` is an error.

E18 If a request for a **specific page** refers to a page that does not exist in the Object, this SHALL be an error. E.g. In an object with only 5 pages, the attribute `Pages="6"` is an error.

E19 No matter how many missing page errors occur in a single `Pages` attribute, only a single error SHALL be returned. E.g. In an object with only 5 pages, the attribute `Pages="3,7,8,2-99,6-"` will only return a single missing pages error.

Example:

An example of a <GetImagePage>element is shown below. This example defines processing to be applied to selected pages from the content of a single retrieved object and *Figure 3.2* shows the structure of this element diagrammatically.

```
<ProcessContent>
    <GetImagePage Pages="1,3,21-45">
        <ManipulateRasterContent>
            <Clip Unit="inches"
                  HeightOrigin="top"
                  HeightStart="0.0"
                  Height="2.27"
                  WidthOrigin="left"
                  WidthStart="2.0"
                  Width="2.33"/>
            <Rotate Degrees="90"/>
        </ManipulateRasterContent>
        <TranscodeContent RequestedContentDescriptor="PNG"/>
    </GetImagePage>
</ProcessContent>
```

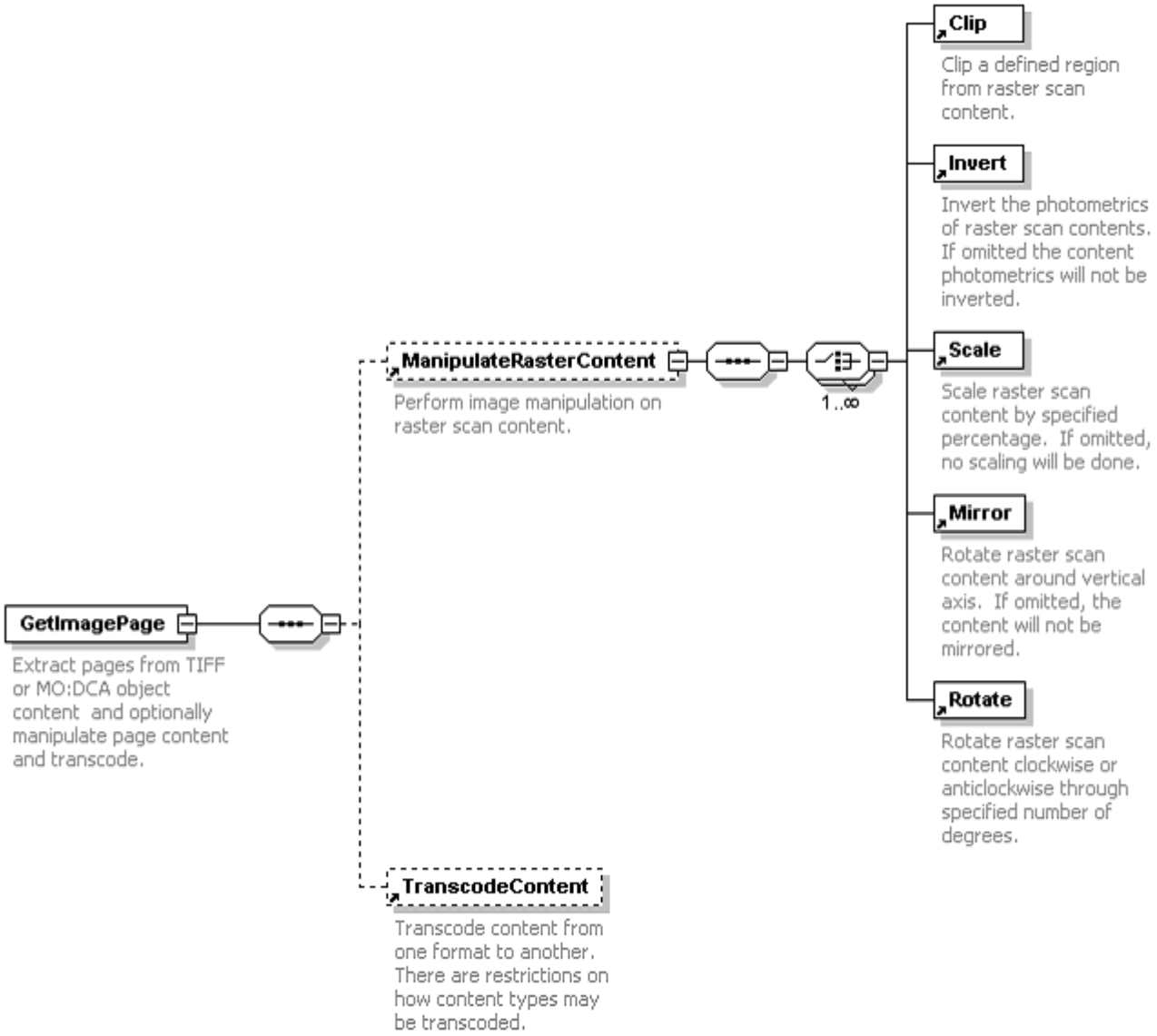


Figure 3-2: Element <GetImagePage> Structure

3.3.7 Check 21 Image Clipping

As a result of the Check 21 legislation, Archive Content Services now provides an option to determine whether a check object is actually a Substitute Check document and, if so, to extract the check image from the document before returning it to the calling application. This feature is provided as a convenience for the application writer, so that the actual check content of a Substitute Check document may easily be extracted and presented to the end user.

E1 Element `<GetImagePage>` in an Archive Content Services request shall have the new optional attributes defined in Table 3 1: Check 21 <GetImagePage> Attributes to let a calling application request that the image content returned for each page be treated as a Substitute Check document.

Name	Value
`SubstituteFieldName`	*Optional.* The name of the object field whose value is used to determine whether the object contains a Substitute Check. The presence of this attribute determines whether the item is to be processed as a Substitute Check.
`SubstituteFieldValues`	*Optional (if present, a `SubstituteFieldName` attribute must also be present).* A comma-separated list of values. If the value of the field with the name defined by attribute `SubstituteFieldName` matches one of these values, the object is assumed to hold a Substitute Check. These values will be matched *exactly* as given against the returned field value. If omitted, the default value of this attribute is "`4`".

Table 3-1: Check 21 `<GetImagePage>` Attributes

An example of a `<GetImagePage>`element with Substitute Check attributes is shown below

```
<ProcessContent>
    <GetImagePage Pages="1f,2b"
                  SubstituteFieldName="EPC"
                  SubstituteFieldValues="4">
        <ManipulateRasterContent>
            <Rotate Degrees="90"/>
        </ManipulateRasterContent>
        <TranscodeContent RequestedContentDescriptor="PNG"/>
    </GetImagePage>
</ProcessContent>
```

E2 If an item is identified as being a Substitute Check, the calling application shall be able to specify, for each page in the image file, whether this page is to be clipped as check front image, as a check back image or not clipped at all. Clipping is to be specified by appending the letter '**`f`**' to the page number in the `Pages` attribute for front image clipping, or the letter '**`b`**' for back image clipping. If neither '`f`' nor '`b`' is present, the image will not be clipped.

For example, the attribute **`Pages="1f,2b"`** will clip the first image page as the front of a check and the second image page as the back of a check.

A3 If an item is not identified as a Substitute Check by having a `SubstituteFieldName` attribute, the presence of any '`f`' and '`b`' page number suffixes shall be ignored and image regions are not to be clipped.

E4 If Substitute Check processing is to be done for an item and that item has a *range* of pages specified in its `Pages` attribute, then both the low and high bounding pages shall have the same suffix. For example, the attribute `Pages="1f-3b"` is invalid. If this happens, an error code is to be returned with no image region..

E5 If an item is identified as being a Substitute Check, the image region that holds the check image shall be clipped from the full Substitute Check image and processed in the same way that the full image would have been processed before being returned in a `<GetImagePageResult>` element. This processing is to include all image manipulations defined by `<Clip>`, `<Invert>`, `<Scale>`, `<Mirror>` and `<Rotate>` elements, as well as any transcoding defined by a `<TranscodeContent>` element.

A6 The location and size of the image regions of a Substitute Check that hold front and rear check images shall be as defined in reference [10], *X9.100-140-2004: Specifications for an Image Replacement Document – IRD*.

E7 When writing out a `<GetImagePageResult>` element for an item where clipping has been specified in a `<GetImagePage>` `Pages` attribute, the '`f`' or '`b`' suffixes shall be removed from the page numbers.

3.4 Archive Result Format

3.4.1 General

C1 The result from a client program request to Archive Content Services SHALL be an Archive Interaction Services XML response document in the format defined in reference [1] *ImageMark Archive Retrieval – Interface Specification – Retrieval Interface*. This XML document may contain ACS elements as defined in this section.

C2 If the response contains object content and the requester has asked for this content to be processed by ACS, the AIS response document SHALL have an additional ACS-specific `<ProcessContentResult>` elements in it as described below. The following AIS services can return object content:

- Inquiry Service – In the `<indexqueryresult>` element from an `<indexquery>` function request.
- Batch Access Service – In the `<indexqueryresult>` child element of the `<getbatchresult>` element from a `<getbatch>` request and in the `<getimageresult>` element from a `<getimage>` request.

A3 In all cases, the `<ProcessContentResult>` element SHALL have the format defined in section 3.4.5, Element <ProcessContentResult> Format.

C4 All elements in XML result documents SHALL be well-formed XML as defined in reference [3] *Extensible Markup Language (XML) 1.0 (Third Edition)*

C5 All binary content SHALL be encoded to the Base64 standard as defined in reference [7], *Multipurpose Internet Mail Extensions(MIME) Part One: Format of Internet Message Bodies*.

C6 If ACS is unable to communicate with AIS for socket failure or any other reason, a simulated AIS error response SHALL be returned to the requester.

3.4.2 Content Processing Results – AIS Inquiry Service

C1 If the response from an AIS Inquiry Service `<indexquery>` function request that has been processed by ACS contains a `<ProcessContentResult>` element, this element SHALL be a child element of the `<item>` element and will displace the `<image>` element that would normally appear here. This is shown in the XML example below.

```
<?xml version="1.0" encoding="UTF-8"?>
<inquiryresult code="0" subcode="0" sessionname="session1">
    <indexqueryresult code="0" subcode="0" queryname="test query 1">
        <item objectname="Commercial" mediastatus="m">
            <rfield name="postingdate" value="20030627"/>
            <rfield name="account_number" value="1234567890"/>
            <ProcessContentResult code=”0” subcode=”0”>
                Processed content
            </ProcessContentResult>
        </item>
        <item objectname="Commercial" mediastatus="m">
            <rfield name="postingdate" value="20021225"/>
            <rfield name="account number" value="0987654321"/>
            <ProcessContentResult code=”0” subcode=”0”>
                Processed content
            </ProcessContentResult>
        </item>
    </indexqueryresult>
</inquiryresult>
```

3.4.3 Content Processing Results – AIS Batch Access Service

C1 If the response from an AIS Batch Access Service `<getbatch>` function request that has been processed by ACS contains a `<ProcessContentResult>` element, this element SHALL be a child element of the `<item>` element and will displace the `<image>` element that would normally appear here. This is shown in the XML example below.

```
<?xml version="1.0" encoding="UTF-8"?>
<batchaccessresult code="0" subcode="0" sessionname="session1">
    <getbatchresult code="0" subcode="0" batchname="test batch 1">
        <indexqueryresult code=”0” subcode=”0” queryname=”testquery 1”>
            <item objectname="Commercial" mediastatus="m">
                <rfield name="postingdate" value="20030627"/>
                <rfield name="account_number" value="1234567890"/>
                <ProcessContentResult code=”0” subcode=”0”>
                    Processed content
```

```
                </ProcessContentResult>
            </item>
            <item objectname="Commercial" mediastatus="m">
                <rfield name="postingdate" value="20021225"/>
                <rfield name="account number" value="0987654321"/>
                <ProcessContentResult code=”0” subcode=”0”>
                    Processed content
                </ProcessContentResult>
            </item>
        </indexqueryresult>
    </getbatchresult>
</batchaccessresult>
```

C2 If the response from an AIS Batch Access Service <getimage> function request that has been processed by ACS contains a <ProcessContentResult> element, this element SHALL be a child element of the <getimageresult> element and will displace the <image> element that would normally appear here. This is shown in the XML example below.

```
<?xml version="1.0" encoding="UTF-8"?>
<batchaccessresult code="0" subcode="0" sessionname="session1">
    <getimageresult code="0" subcode="0" batchname="test batch 1">
        <ProcessContentResult code=”0” subcode=”0”>
            Processed content
        </ProcessContentResult>
        <ProcessContentResult code=”0” subcode=”0”>
            …
        </ProcessContentResult>
    </getimageresult>
</batchaccessresult>
```

3.4.4 Unrecoverable Errors

C1 If an unrecoverable error occurs at any point during processing, a <TerminatedResponse> element SHALL be written to the XML result at the point where the error occurred. All open elements in the request are then to have their closing tags written to the XML stream to ensure that a valid and well-formed XML document is returned to the requesting application. An example of the use of a <TerminatedResponse> element can be seen in the XML example below, which shows the case where AIS has failed while returning the result of an Inquiry Service request.

```
<?xml version="1.0" encoding="UTF-8"?>
<inquiryresult code="0" subcode="0" sessionname="session1">
    <indexqueryresult code="0" subcode="0" queryname="test query 1">
        <item objectname="Commercial" mediastatus="m">
            <rfield name="postingdate" value="20030627"/>
            <rfield name="account_number" value="1234567890"/>
            Something bad happened to AIS here
            <TerminatedResponse code="13" subcode="5"/>
        </item>
    </indexqueryresult>
</inquiryresult>
```

3.4.5 Element <ProcessContentResult> Format

For this release of Archive Content Services, manipulation and transcoding of object content will only be available for NCR TIFF and MO:DCA objects through the <GetImagePage> element that returns a <GetImagePageResult> element. Section 3.4.5 has been left in the specification because it is anticipated that the more general ability to manipulate and transcode other object content will implemented in later releases.

A1 The <ProcessContentResult> element SHALL contain object content that has been processed as the result of an ACS request.

A2 The format of the <ProcessContentResult> element SHALL be as defined in the XML Schema that is documented in section 5, *Appendix B: Element <ProcessContentResult> XML Schema*.

E3 If AIS fails to return a valid image for any reason, element <ProcessContentResult> SHALL have code and subcode attributes indicating the cause of the error. In this situation an <image> child element will not be present.

An example of a <ProcessContentResult> element is shown below. This example shows the results from processing the content of a single retrieved object without any content extraction. Figure 3 3 shows the structure of this element diagrammatically.

```
<ProcessContentResult code="0" subcode="0">
    <image status="0" original_len="12345" encoded_len="23456">
        encoded content data from AIS - transcoded and/or manipulated
    </image>
</ProcessContentResult>
```

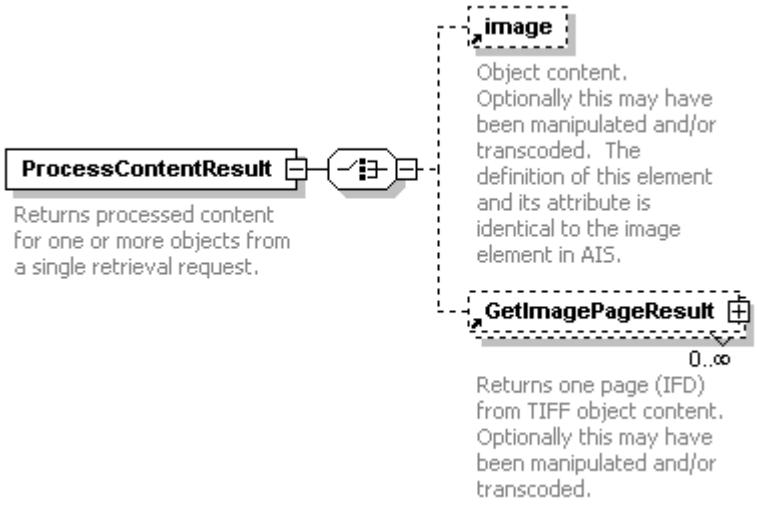


Figure 3-3: Element <ProcessContentResult> Structure

3.4.6 Element <GetTiffPageResult> Format

E1 For back-compatibility reasons, element `<GetTiffPageResult>` shall be retained for cases where a request has been made using a `<GetTiffPage>` element. However, it is to have exactly the same format as element `<GetImagePageResult>` as defined in section 3.4.7, Element <GetImagePageResult> Format.

Note: *All other requirements previously defined in this section have now been moved to section* 3.4.7, Element <GetImagePageResult> Format.

3.4.7 Element <GetImagePageResult> Format

E1 A `<GetImagePageResult>` element SHALL return the results of extracting one page from the content of one multi-page TIFF or MO:DCA object. Each page may optionally have been manipulated and/or transcoded.

E2 If the `<GetImagePage>` request element requests more than one page, there SHALL be a separate `<GetImagePageResult>` element for each page returned.

E3 The format of a `<GetImagePageResult>` element SHALL be as defined in the XML Schema that is documented in section. 5, Appendix B: Element <ProcessContentResult> XML Schema

An example of a `<GetImagePageResult>` element is shown below and Figure 3 4 below shows the structure of the element diagrammatically.

```
<GetImagePageResult code="0" subcode="0" PageNumber="1">
    <image status="0" original_len="4800" encoded_len="6400" encode_type="base64">
        encoded content data from AIS for page - transcoded and/or manipulated
    </image>
</GetImagePageResult>
```

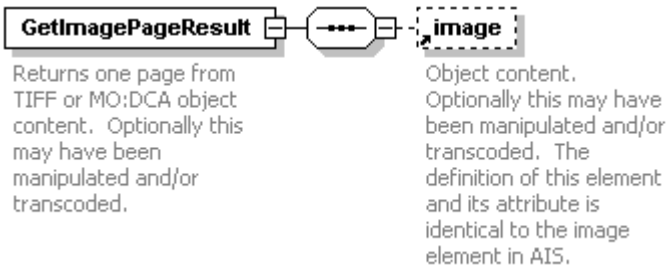


Figure 3-4: Element `<GetImagePageResult>` Structure

4. Appendix A: Element <ProcessContent> XML Schema

Schema **ProcessContent.xsd**

element **Clip**

diagram

used by element **ManipulateRasterContent**

attributes

Name	Type	Use	Default	Fixed	Annotation
Unit	xs:string	optional	pixels		The unit used in defining positions and dimensions. If this unit is "pixels", attributes "HeightStart", "Height", "WidthStart" and "Width" must be non-negative integers. If "inches" then non-negative decimal fractions may be used.
HeightOrigin	xs:string	optional	top		Whether the top or bottom edge of the image is used as origin for height start and height values.
HeightStart	xs:float	optional	0		The row where the clip region starts in the specified unit. Must be non-negative and within the boundary of the image.
Height	xs:float	optional	0		The height of the clip region in the specified unit. May be positive, negative or zero. If zero, clip will be extended in positive direction to image edge. If extends beyond image edge, clip will be truncated at image edge. If negative, clip will begin at HeightStart but will extend back in negative direction towards HeightOrigin.
WidthOrigin	xs:string	optional	left		Whether the left or right edge of the image is used as origin for width start and width values.
WidthStart	xs:float	optional	0		The column where the clip region starts in the specified unit. Must be non-negative and within the boundary of the image.
Width	xs:float	optional	0		The width of the clip region in the specified unit. May be positive, negative or zero. If zero, clip will be extended in positive direction to image edge. If extends beyond image edge, clip will be truncated at image edge. If negative, clip will begin at WidthStart but will extend back in negative direction towards WidthOrigin.

annotation documentation Clip a defined region from raster scan content.

source

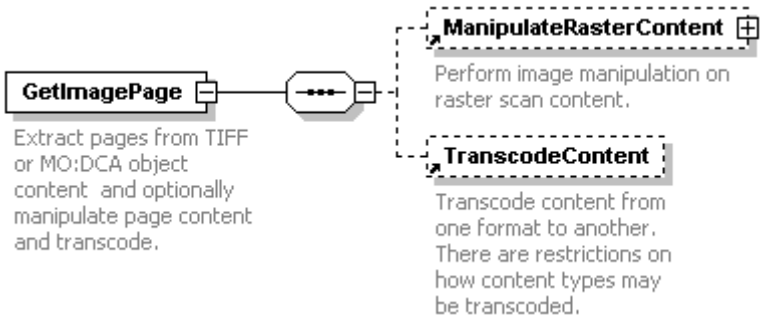
```
<xs:element name="Clip">
 <xs:annotation>
  <xs:documentation>Clip a defined region from raster scan content.</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:attribute name="Unit" use="optional" default="pixels">
```

```
      <xs:annotation>
        <xs:documentation>The unit used in defining positions and dimensions.  If this unit is "pixels", attributes "HeightStart", "Height", "WidthStart" and "Width" must
be non-negative integers.  If "inches" then non-negative decimal fractions may be used.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="pixels"/>
          <xs:enumeration value="inches"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="HeightOrigin" use="optional" default="top">
      <xs:annotation>
        <xs:documentation>Whether the top or bottom edge of the image is used as origin for height start and height values.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="top"/>
          <xs:enumeration value="bottom"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="HeightStart" type="xs:float" use="optional" default="0">
      <xs:annotation>
        <xs:documentation>The row where the clip region starts in the specified unit.  Must be non-negative and within the boundary of the image.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="Height" type="xs:float" use="optional" default="0">
      <xs:annotation>
        <xs:documentation> The height of the clip region in the specified unit.  May be positive, negative or zero.  If zero, clip will be extended in positive direction to
image edge.  If extends beyond image edge, clip will be truncated at image edge.  If negative, clip will begin at HeightStart but will extend back in negative direction
towards HeightOrigin.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="WidthOrigin" use="optional" default="left">
      <xs:annotation>
        <xs:documentation>Whether the left or right edge of the image is used as  origin for width start and width values.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="left"/>
          <xs:enumeration value="right"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="WidthStart" type="xs:float" use="optional" default="0">
      <xs:annotation>
        <xs:documentation>The column where the clip region starts in the specified unit. Must be non-negative and within the boundary of the
image.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="Width" type="xs:float" use="optional" default="0">
      <xs:annotation>
        <xs:documentation> The width of the clip region in the specified unit.  May be positive, negative or zero.  If zero, clip will be extended in positive direction to
image edge.  If extends beyond image edge, clip will be truncated at image edge.  If negative, clip will begin at WidthStart but will extend back in negative direction
towards WidthOrigin.</xs:documentation>
```

```
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

element **GetImagePage**

diagram



children **ManipulateRasterContent** **TranscodeContent**

used by element **ProcessContent**

attributes

Name	Type	Use	Default	Fixed	Annotation
Pages	xs:string	optional			Pages to be extracted from TIFF or MO:DCA object content. Enter page numbers and/or pages ranges separated by commas. For example: "1,3,5-12". If a page range does not have an upper limit; for example "2-" the upper limit is the final page. . It is not an error to request the same page more than once.
SubstituteFieldName	xs:string	optional			The name of the object field whose value is used to determine whether the object contains a Substitute Check. The presence of this attribute determines whether the item is to be processed as a Substitute Check
SubstituteFieldValues	xs:string	optional	4		A comma-separated list of values. If the value of the field with the name defined by attribute SubstituteFieldName matches one of these values, the object is assumed to hold a Substitute Check. These values will be matched *exactly* as given against the returned field value.

annotation documentation Extract pages from TIFF or MO:DCA object content and optionally manipulate page content and transcode.

source

```
<xs:element name="GetImagePage">
 <xs:annotation>
  <xs:documentation>Extract pages  from TIFF or MO:DCA object content  and optionally manipulate page content and transcode.</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:sequence>
   <xs:element ref="ManipulateRasterContent" minOccurs="0"/>
   <xs:element ref="TranscodeContent" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="Pages" type="xs:string" use="optional">
   <xs:annotation>
    <xs:documentation>Pages to be extracted from object content.  Enter page numbers and/or pages ranges separated by commas.  For example: "1,3,5-12".  If a
page range does not have an upper limit; for example "2-" the upper limit is the final page.  .  It is not an error to request the same page more than
once.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="SubstituteFieldName" type="xs:string" use="optional">
   <xs:annotation>
```

```
      <xs:documentation>The name of the object field whose value is used to determine whether the object contains a Substitute Check.  The presence of this
attribute determines whether the item is to be processed as a Substitute Check</xs:documentation>
    </xs:annotation>
   </xs:attribute>
   <xs:attribute name="SubstituteFieldValues" type="xs:string" use="optional" default="4">
    <xs:annotation>
     <xs:documentation>A comma-separated list of values.  If the value of the field with the name defined by attribute SubstituteFieldName matches one of these
values, the object is assumed to hold a Substitute Check.  These values will be matched exactly as given against the returned field value.</xs:documentation>
    </xs:annotation>
   </xs:attribute>
  </xs:complexType>
</xs:element>
```

element **Invert**

diagram

used by element **ManipulateRasterContent**

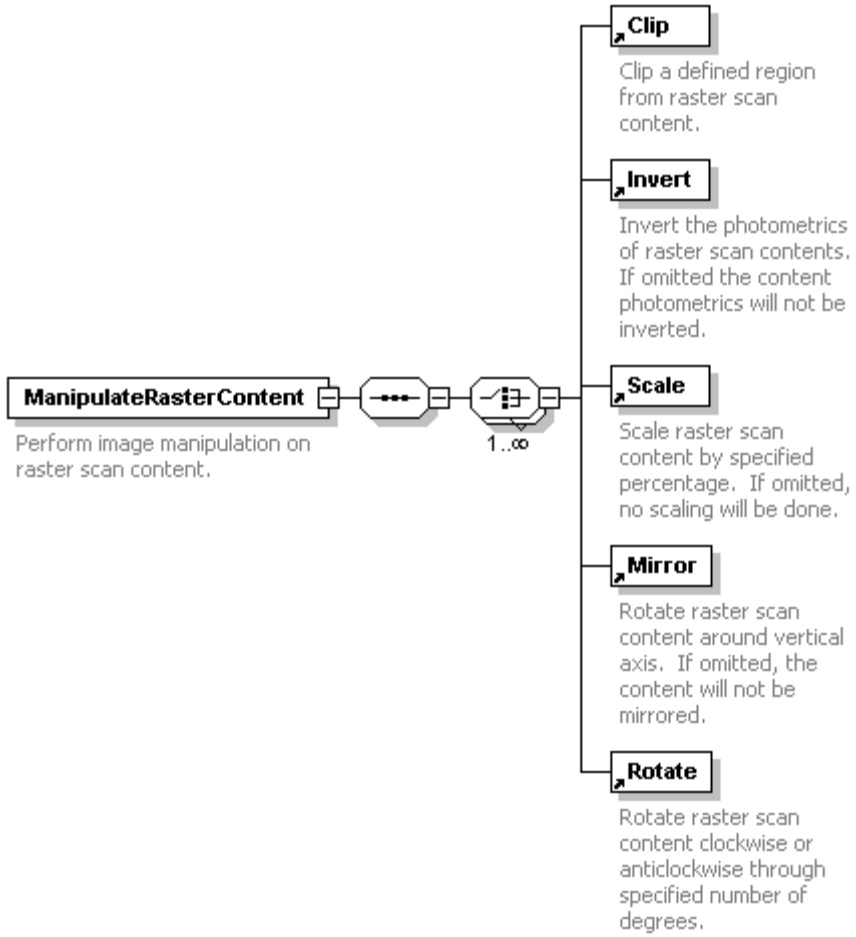
annotation documentation Invert the photometrics of raster scan contents. If omitted the content photometrics will not be inverted.

source

```
<xs:element name="Invert" default="0">
 <xs:annotation>
  <xs:documentation>Invert the photometrics of raster scan contents.  If omitted the content photometrics will not be inverted.</xs:documentation>
 </xs:annotation>
</xs:element>
```

element **ManipulateRasterContent**

diagram



children **Clip** **Invert** **Scale** **Mirror** **Rotate**

used by
elements **GetImagePage** **ProcessContent**

annotation
documentation Perform image manipulation on raster scan content.

source

```
<xs:element name="ManipulateRasterContent">
  <xs:annotation>
    <xs:documentation>Perform image manipulation on raster scan content.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
```

```
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="Clip"/>
        <xs:element ref="Invert"/>
        <xs:element ref="Scale"/>
        <xs:element ref="Mirror"/>
        <xs:element ref="Rotate"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **Mirror**

diagram

used by element **ManipulateRasterContent**

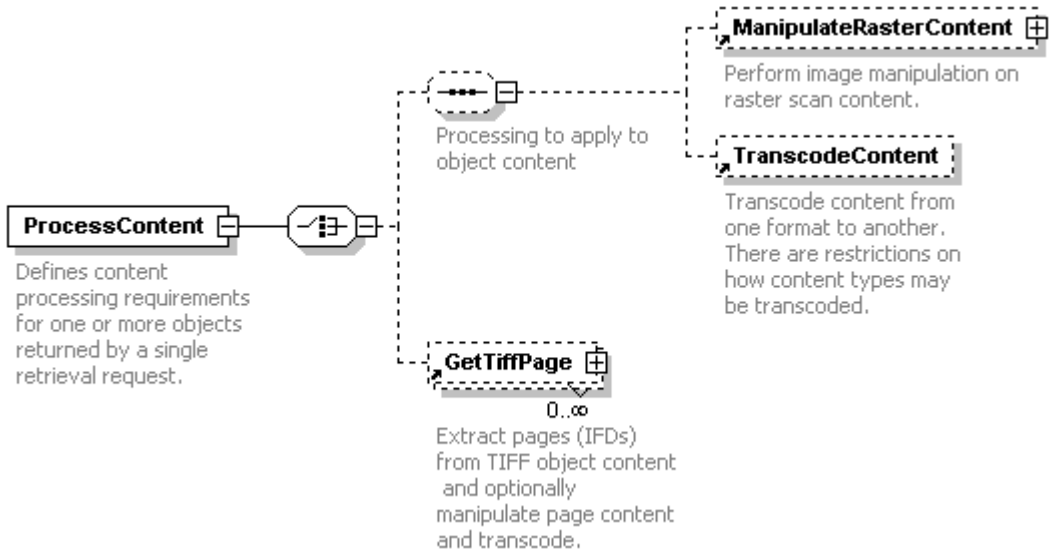
annotation documentation Rotate raster scan content around vertical axis. If omitted, the content will not be mirrored.

source

```
<xs:element name="Mirror" default="0">
 <xs:annotation>
  <xs:documentation>Rotate raster scan content around vertical axis.  If omitted, the content will not be mirrored.</xs:documentation>
 </xs:annotation>
 <xs:complexType/>
</xs:element>
```

element **ProcessContent**

diagram



children **ManipulateRasterContent** **TranscodeContent** **GetImagePage**

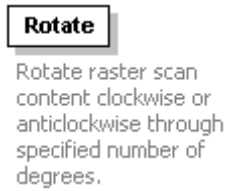
annotation documentation Defines content processing requirements for one or more objects returned by a single retrieval request.

source

```
<xs:element name="ProcessContent">
 <xs:annotation>
  <xs:documentation>Defines content processing requirements for one or more objects returned by a single retrieval request.</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:choice>
   <xs:sequence minOccurs="0">
    <xs:annotation>
     <xs:documentation>Processing to apply to object content </xs:documentation>
    </xs:annotation>
    <xs:element ref="ManipulateRasterContent" minOccurs="0"/>
    <xs:element ref="TranscodeContent" minOccurs="0"/>
   </xs:sequence>
   <xs:element ref="GetImagePage" minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>
 </xs:complexType>
</xs:element>
```

element **Rotate**

diagram



used by element **ManipulateRasterContent**

attributes

Name	Type	Use	Default	Fixed	Annotation
Degree	xs:int	required			Number of degrees to rotate raster scan image. This may be positive for clockwise rotation or negative for anticlockwise rotation and must be ±90, ±180 or ±270.

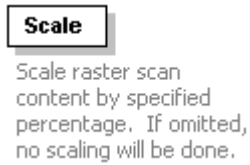
annotation documentation Rotate raster scan content clockwise or anticlockwise through specified number of degrees.

source

```
<xs:element name="Rotate">
 <xs:annotation>
  <xs:documentation>Rotate raster scan content clockwise or anticlockwise through specified number of degrees.</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:attribute name="Degree" type="xs:int" use="required">
   <xs:annotation>
    <xs:documentation>Number of degrees to rotate raster scan image.  This may be positive for clockwise rotation or negative for anticlockwise rotation and must be a multiple of 90.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
 </xs:complexType>
</xs:element>
```

element **Scale**

diagram



used by element **ManipulateRasterContent**

attributes

Name	Type	Use	Default	Fixed	Annotation
Factor	xs:int	required			Percentage scale factor to be applied to raster scan image. Factors of less than 100 will result in a reduction and more than 100 will give a magnification.

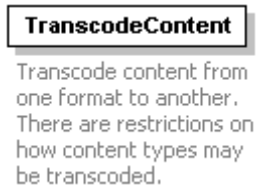
annotation documentation Scale raster scan content by specified percentage. If omitted, no scaling will be done.

source

```
<xs:element name="Scale">
 <xs:annotation>
  <xs:documentation>Scale raster scan content by specified percentage.  If omitted, no scaling will be done. </xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:attribute name="Factor" use="required">
   <xs:annotation>
    <xs:documentation>Percentage scale factor to be applied to raster scan image.  Factors of less than 100 will result in a reduction and more than 100 will give a
magnification.</xs:documentation>
   </xs:annotation>
   <xs:simpleType>
    <xs:restriction base="xs:int">
     <xs:minInclusive value="1"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
</xs:element>
```

element **TranscodeContent**

diagram



used by
elements **GetImagePage**
ProcessContent

attributes

Name	Type	Use	Default	Fixed	Annotation
RequestedContentDescriptor	xs:string	required			The content type that the object type is to be transcoded into.

annotation
documentation Transcode content from one format to another. There are restrictions on how content types may be transcoded.

source

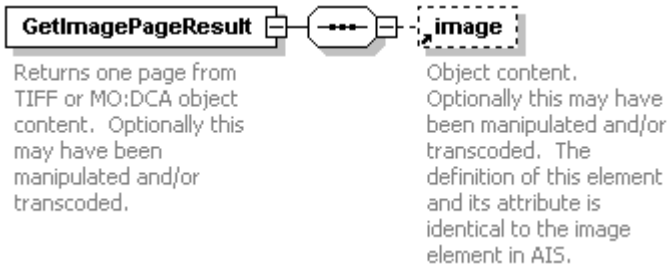
```
<xs:element name="TranscodeContent">
 <xs:annotation>
  <xs:documentation>Transcode content from one format to another.  There are restrictions on how content types may be transcoded.</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:attribute name="RequestedContentDescriptor" use="required">
   <xs:annotation>
    <xs:documentation>The content type that the object type is to be transcoded into.</xs:documentation>
   </xs:annotation>
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:enumeration value="PNG"/>
     <xs:enumeration value="JPG"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
</xs:element>
```

5. Appendix B: Element <ProcessContentResult> XML Schema

Schema **ProcessContentResult.xsd**

element **GetImagePageResult**

diagram



children **image**

used by element **ProcessContentResult**

attributes

Name	Type	Use	Default	Fixed	Annotation
code	xs:int	required			
subcode	xs:int	required			
PageNumber	xs:int	optional			Number of the page extracted from TIFFor MO:DCA object content. Each extracted TIFF page will have a separate GetImagePageResult element. In the event of an error due to one or more non-existent pages being requested, this attribute will be omitted from the response element.
status	xs:int	optional			Indicates why an image was not returned. Can be: 1 = image on tape; 2 = image aged out; 3 = RAID down.

annotation

documentation Returns one page from TIFF or MO:DCA object content. Optionally this may have been manipulated and/or transcoded.

source

```
<xs:element name="GetImagePageResult">
 <xs:annotation>
  <xs:documentation>Returns one page from TIFF or MO:DCA object content.  Optionally this may have been manipulated and/or transcoded.</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:sequence>
   <xs:element ref="image" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="code" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:int">
     <xs:minInclusive value="0"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="subcode" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:int">
```

```
          <xs:minInclusive value="0"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="PageNumber" use="required">
      <xs:annotation>
        <xs:documentation>Number of the page extracted from object content.  Each extracted page will have a separate GetImagePageResult
element.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:minInclusive value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="status" use="optional">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:minInclusive value="0"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

element **image**

diagram

used by
elements **GetImagePageResult**
ProcessContentResult

attributes

Name	Type	Use	Default	Fixed	Annotation
status	xs:int	required			
original_len	xs:int	optional			
encoded_len	xs:int	optional			
encode_type	xs:string	optional		base64	

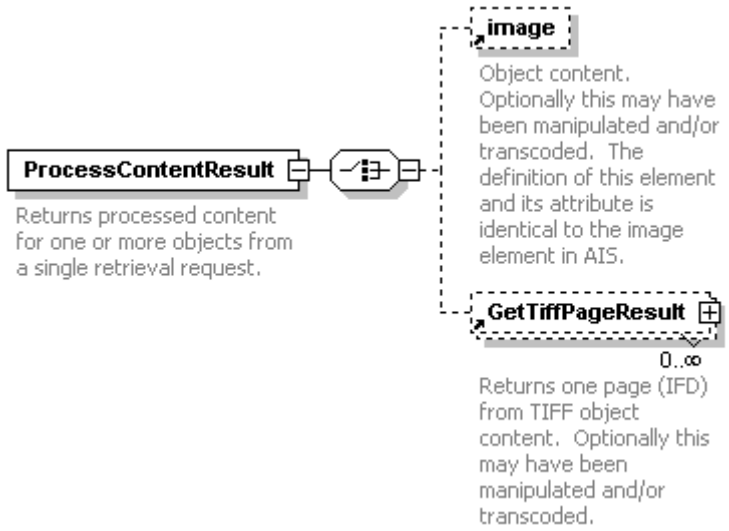
annotation
documentation Object content. Optionally this may have been manipulated and/or transcoded. The definition of this element and its attribute is identical to the image element in AIS.

source

```
<xs:element name="image">
 <xs:annotation>
  <xs:documentation>Object content.  Optionally this may have been manipulated and/or transcoded.  The definition of this element and its attribute is identical to the image element in AIS.</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:attribute name="status" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:int">
     <xs:minInclusive value="0"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="original_len" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:int">
     <xs:minInclusive value="1"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="encoded_len" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:int">
     <xs:minInclusive value="1"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="encode_type" type="xs:string" use="optional" fixed="base64"/>
 </xs:complexType>
</xs:element>
```

element **ProcessContentResult**

diagram



children **image** **GetImagePageResult**

attributes

Name	Type	Use	Default	Fixed	Annotation
code	xs:int	required			
subcode	xs:int	required			

annotation
documentation Returns processed content for one or more objects from a single retrieval request.

source

```
<xs:element name="ProcessContentResult">
 <xs:annotation>
  <xs:documentation>Returns processed content for one or more objects from a single retrieval request.</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:choice>
   <xs:element ref="image" minOccurs="0"/>
   <xs:element ref="GetImagePageResult" minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>
  <xs:attribute name="code" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:int">
     <xs:minInclusive value="0"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="subcode" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:int">
     <xs:minInclusive value="0"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
```

```
</xs:element>
```

6. Appendix C: Return Codes and Subcodes

This section lists the return codes and subcodes that may be returned by ACS in the `<ProcessContentResult>` and `<GetImagePageResult>` elements.

6.1 Codes Ordered by Reason

6.1.1 General Codes

Code	Subcode	Reason
0	0	Success
13	2	AIS is not available
13	4	Unable to submit complete request to AIS
13	8	Initialization of ACS service failed

6.1.2 Codes Due to Invalid Requests

Code	Subcode	Reason
10	1	ProcessContent element – invalid child element
11	1	GetImagePage element – invalid child element
11	7	GetImagePage element – Pages attribute has invalid range specification
11	8	GetImagePage element – Pages attribute requests non-existent page number.
11	12	GetImagePage element – requires SubstituteFieldName attribute because SubstituteFieldValues are defined
12	1	ManipulateRasterContent element – invalid child element
12	10	Clip element – invalid child element.
12	11	Clip element – Unit attribute has an invalid value.
12	12	Clip element – HeightOrigin attribute has an invalid value.
12	13	Clip element – HeightStart attribute has an invalid value.
12	14	Clip element – Height attribute has an invalid value.
12	15	Clip element – WidthOrigin attribute has an invalid value.
12	16	Clip element – WidthStart attribute has an invalid value.
12	17	Clip element – Width attribute has an invalid value.
12	20	Rotate element – invalid child element.
12	21	Rotate element – requires Degree attribute.
12	22	Rotate element – Degree attribute has an invalid value.
12	30	Scale element – invalid child element.
12	31	Scale element – requires Factor attribute.
12	32	Scale element – Factor attribute has an invalid value.
12	40	Invert element – invalid child element.

Code	Subcode	Reason
12	50	Mirror element – invalid child element
12	60	TranscodeContent element – invalid child element
12	61	TranscodeContent – requires RequestedContentDescriptor attribute
12	62	TranscodeContent element – RequestedContentDescriptor attribute has an invalid value
12	70	Fill element has an invalid child element
12	71	Fill element's Unit attribute has an invalid value
12	72	Fill element's HeightOrigin attribute has an invalid value
12	73	Fill element's HeightStart attribute has an invalid value
12	74	Fill element's Height attribute has an invalid value
12	75	Fill element's WidthOrigin attribute has an invalid value
12	76	Fill element's WidthStart attribute has an invalid value
12	77	Fill element's Width attribute has an invalid value
12	78	Fill element's Color attribute has an invalid value
13	3	Request is an invalid XML document.

6.1.3 Codes Due to AIS Responses

Code	Subcode	Reason
10	2	ProcessContentResult element – image not retrieved.
10	3	ProcessContentResult element – image retrieved as an image handle.
10	4	ProcessContentResult element – image has length 0.
10	5	ProcessContentResult element – image element does not have required attributes or has invalid attribute values.
10	6	ProcessContentResult element – encoded data is not valid.
10	7	ProcessContentResult element – unable to manipulate/transcode requested page.
11	2	GetImagePageResult element – image not retrieved.
11	3	GetImagePageResult element – image retrieved as an image handle.
11	4	GetImagePageResult element – image has length 0.
11	6	GetImagePageResult element –encoded data is not valid.
11	9	GetImagePageResult element – unable to extract requested page.
11	10	GetImagePageResult element – unable to manipulate/transcode requested page.
11	11	GetImagePageResult element – image not a valid TIFF or MO:DCA file.

6.1.4 Codes Due to Unrecoverable Errors

These codes may appear anywhere inside an XML result document and are due to ACS being unable to continue processing the result due to an incorrect or incomplete response or other unrecoverable error. After writing out a `<TerminatedResponse>`,

ACS will write out closing tags of all open XML elements in the correct order. This allows the requesting client that receives the response to parse it as a well-formed (if unsatisfactory) XML document.

Code	Subcode	Reason
13	1	TerminatedResponse element – unexpected application error
13	5	TerminatedResponse element – response from AIS is invalid XML document
13	6	TerminatedResponse element – unable to retrieve complete response from AIS
13	7	TerminatedResponse element – unable to log ARS message and ARS fail behavior is to terminate

6.2 Codes in Numeric Order

Code	Subcode	Reason
0	0	Success
10	1	ProcessContent element – invalid child element
10	2	ProcessContentResult element – image not retrieved.
10	3	ProcessContentResult element – image retrieved as an image handle.
10	4	ProcessContentResult element – image has length 0.
10	5	ProcessContentResult element – image element does not have required attributes or has invalid attribute values.
10	6	ProcessContentResult element – encoded data is not valid.
10	7	ProcessContentResult element – unable to manipulate/transcode requested page.
11	1	GetImagePage element – invalid child element
11	2	GetImagePageResult element – image not retrieved.
11	3	GetImagePageResult element – image retrieved as an image handle.
11	4	GetImagePageResult element – image has length 0.
11	6	GetImagePageResult element –encoded data is not valid.
11	7	GetImagePage element – Pages attribute has invalid range specification
11	8	GetImagePage element – Pages attribute requests non-existent page number.
11	9	GetImagePageResult element – unable to extract requested page.
11	10	GetImagePageResult element – unable to manipulate/transcode requested page.
11	11	GetImagePageResult element – image not a valid TIFF or MO:DCA file.
11	12	GetImagePage element – requires SubstituteFieldName attribute because SubstituteFieldValues are defined
12	1	ManipulateRasterContent element – invalid child element
12	10	Clip element – invalid child element.
12	11	Clip element – Unit attribute has an invalid value.
12	12	Clip element – HeightOrigin attribute has an invalid value.
12	13	Clip element – HeightStart attribute has an invalid value.
12	14	Clip element – Height attribute has an invalid value.
12	15	Clip element – WidthOrigin attribute has an invalid value.
12	16	Clip element – WidthStart attribute has an invalid value.
12	17	Clip element – Width attribute has an invalid value.
12	20	Rotate element – invalid child element.
12	21	Rotate element – requires Degree attribute.
12	22	Rotate element – Degree attribute has an invalid value.
12	30	Scale element – invalid child element.
12	31	Scale element – requires Factor attribute.

Code	Subcode	Reason
12	32	Scale element – Factor attribute has an invalid value.
12	40	Invert element – invalid child element.
12	50	Mirror element – invalid child element
12	60	TranscodeContent element – invalid child element
12	61	TranscodeContent – requires RequestedContentDescriptor attribute
12	62	TranscodeContent element – RequestedContentDescriptor attribute has an invalid value
13	1	TerminatedResponse element – unexpected application error
13	2	AIS is not available
13	3	Request is an invalid XML document.
13	4	Unable to submit complete request to AIS
13	5	TerminatedResponse element – response from AIS is invalid XML document
13	6	TerminatedResponse element – unable to retrieve complete response from AIS
13	7	TerminatedResponse element – unable to log ARS message and ARS fail behavior is to terminate
13	8	Initialization of ACS service failed

7. Appendix D: Codes/Subcodes for Interaction Services

Each service has a dedicated range of error codes. The error code is returned in two parts, a "code" describing the major status, and a "subcode" describing the minor status. A "code=0" always means successful completion.

The ranges allocated are as follows:

Service	Code	Subcode
Generic (covers all)	1-9	1-100
RESERVED for ACS (Archive Content Services)	10-19	1-xxx (as determined by ACS)
Login	20-29	1-100
Security	30-49	1-100
Inquiry	50-69	1--- xxx (returned by Oracle)
Batch access	70-79	1-100
Update	80-89	1-100

7.1 Codes/Subcodes for generic use by all services

Code	Subcode	Service/Function	Description
0	0	All	Request successful.
1	1	All	Request syntax error.
1	2	All	Sectoken invalid. Service denied.
1	3	All	No permissions. Service denied.
1	4	All	Invalid Service.
1	5	All	Invalid session. "nosession" is returned in the response for the sessionname value.
2	1	All	Interaction Service Down. Service not available.
3	1	All	Database access error.
3	2	All	Invalid application name.

7.2 Codes/Subcodes for Login Manager Service

Code	Subcode	Service/Function	Description
0	0	All	Request successful
0	nn	login	Login successful. "nn" days left until expiry
21	1	login	Login failed. (Userid/password wrong).
21	4	login	Already logged in
21	5	login	System resources unavailable. Fails to allocate a security object.
21	6	login	Unable to locate pwd_encoding attribute. The user is not configured properly.
21	7	login	System error - Security object initialization failure
21	8	login	System error - No user attributes found in database
22	2	login	Too many attempts to log in
22	3	login	Too long since last password change

7.3 Codes/Subcodes for Security Service

Code	Subcode	Service/Function	Description
0	0	All	Successful completion.
0	-1	adduser	Login successful, password is set to "never expire".
30	1	security service	No privileges
30	2	security service	Group does not exist
30	3	security service	User does not exist
30	4	security service	Attribute does not exist
30	5	security service	Invalid character in attribute value
30	6	security service	Domain does not exist
32	1	addgroup	Bad group name

Code	Subcode	Service/Function	Description
32	2	addgroup	Duplicate group name in the domain
32	3	addgroup	Cannot create new domain at this level
32	4	addgroup	Parent group does not exist in the domain
34	1	changegroup	New parent group does not exist in the domain
34	2	changegroup	New group already exists in domain
35	1	deletegroup	Group is not empty
36	1	adduser	Invalid user name
36	2	adduser	Unsupported password encoding
36	3	adduser	Can_grant must be either “Y” or “N”
36	4	adduser	Can_change_pwd must be either “Y” or “N”
36	5	adduser	Change period less than 0 or greater than 365
36	6	adduser	Max_attempts less than 1 or greater than 5
36	7	adduser	Timeout less than 1 or greater than 1440
36	8	adduser	Administrator must be “Y” or “N”
36	9	adduser	Multiuse must be “Y” or “N”
36	10	adduser	Duplicate user name. The User already exists.
38	1	getuser	User is not defined. No group attribute.
39	1	changeuser	User is not defined or invalid No group attribute.
39	2	changeuser	Duplicate user name and domain combination. The user already exists.
39	3	changeuser	Can_grant must be either “Y” or “N”
39	4	changeuser	Can_change_pwd must be either “Y” or “N”
39	5	changeuser	Change period less than 0 or greater than 365
39	6	changeuser	Max_attempts less than 1 or greater than 5
39	7	changeuser	Timeout less than 1 or greater than 1440
39	8	changeuser	Administrator must be “Y” or “N”
39	9	changeuser	Multiuse must be “Y” or “N”
39	10	Changeuser	Unsupported password encoding
39	11	changeuser	Password change failed. New password cannot be the same as existing password.
40	1	deleteuser	User is not defined. No group attribute.
41	1	addattribute	The group does not exist
42	2	changeattribute	The user does not exist

7.4 Codes/Subcodes for Inquiry Service

Code	Subcode	Service/Function	Description
0	0	All	Request successful.
50	2	inquiry service	Invalid syntax
50	3	inquiry service	Invalid attributes
50	4	inquiry service	Batch file already exists
50	5	inquiry service	Could not create batch file
51	1	querycapabilities	Reserved
52	1	indexquery	Invalid syntax
52	2	indexquery	Invalid attributes
52	3	indexquery	Query too big. Aborted.
52	4	indexquery	Posting Date is missing or the range exceeds the configured maximum range
52	5	indexquery	Request spans different databases
52	6	indexquery	Invalid database reference
52	7	indexquery	Error in creating SQL statement
52	8	indexquery	cannot get Data Base Provider (DBP)
52	9	indexquery	Reserved
52	10	indexquery	Invalid sortfield: sortfield must be one of the resultfields.
52	11	indexquery	Must be only one objecthandle field in set of qfields
52	12	indexquery	Low and high values of the objecthandle field must be equal
52	13	indexquery	Invalid objecttype reference

Code	Subcode	Service/Function	Description
53	xxx	indexquery	Subcode contains SQL code
54	2	indexquery (with update)	Invalid updates. Trying to update an inactive table. Resubmit query without updates.
54	3	indexquery (with update)	Invalid updates. Updates not allowed when used "count" request. Resubmit query without updates.
54	4	indexquery (with update)	No permissions to update. The can_update or audit_update flags are not set for these ufields.
54	5	indexquery (with update)	Actual no of items exceed maxrows. No updates done.

7.5 Codes/Subcodes for Batch Access Service

Code	Subcode	Service/Function	Description
0	0	All	Request successful.
70	2	batch service	Invalid syntax or attributes
71	1	cancel, list, delete	no permissions
71	2	cancel, list, delete, get batch	Batch does not exist
71	3	delete, get batch	Invalid state
71	4	getimage	Invalid imagehandle
71	5	All	Mandatory field required
71	99	All	Any other error. See error log for details.

7.6 Codes/Subcodes for Update Service

Code	Subcode	Service/Function	Description
0	0	All	Request successful.
0	nn	indexupdate	Non-error returns from indexupdate have code=0, but subcode equal to the number of rows affected.
80	1	update service	Internal error in update service
80	2	update service	Invalid syntax
80	4	all update functions	Batch file already exists
80	5	all update functions	Could not create batch file
80	6	all update functions	Cannot create offlinerequestdir for batch file
80	7	all update functions	Batch file.oef already exists
80	8	all update functions	Could not create batch file.oef
81	1	querycapabilities	Cannon get Querycapabilities
82	1	indexupdate	IndexUpdate Can_query is not set for this qfield
82	2	indexupdate	IndexUpdate Can_update is not set for this ufield
82	3	indexupdate	IndexUpdate Can_update or part of index is not set for this cfield
82	5	indexupdate	IndexUpdate Update type is copy-update, but no cfield elements specified
82	6	indexupdate	IndexUpdate Unknown duplicates flag
82	7	indexupdate	IndexUpdate Unknown update_type
82	8	indexupdate	IndexUpdate Objectname is not defined
82	9	indexupdate	IndexUpdate Objectname does not exist
82	10	indexupdate	IndexUpdate Postingdate field does not exist or has invalid range.
82	12	indexupdate	IndexUpdate There is no database information for that object name
82	14	indexupdate	IndexUpdate Field does not exist in requested table
82	15	indexupdate	IndexUpdate Record does not exist
82	16	indexupdate	IndexUpdate Duplicates flag set to NO
82	19	indexupdate	There are not 'Update' fields.
82	20	indexupdate	Destination posting date is not in active table.
82	21	indexupdate	There are no 'Copy' fields.
82	22	indexupdate	Posting date is already exists.
82	23	indexupdate	Field presents in ufield and cfield
82	24	indexupdate	Cannot change posting date because the item is already in inactive table
82	25	indexupdate	Cannot create select list
82	26	indexupdate	Cannot create where list

Code	Subcode	Service/Function	Description
82	27	indexupdate	Prevent posting dates to be processed at the same time
82	28	indexupdate	IndexUpdate: updatename is not defined
82	29	indexupdate	IndexUpdate: updatetype is not defined
82	30	indexupdate	IndexUpdate: Missing value for cfield
82	31	indexupdate	IndexUpdate: Missing low and high attributes for qfield
82	32	indexupdate	IndexUpdate: Missing value for ufield
82	33	indexupdate	IndexUpdate: duplicates is not defined
82	34	indexupdate	Item with this unique key already exists
82	35	indexupdate	Must be only one objecthandle field in set of qfields
82	36	indexupdate	Low and high values of the objecthandle field must be equal
82	37	indexupdate	Invalid objecttype reference
83	xxx	indexupdate	IndexUpdate Subcode contains SQL code

8. Appendix E: Object Content Operations

Figure 8-1, below shows how stored object content may be transcoded into presented content in ACS. Transcoding is only done when requested.

For this release of Archive Content Services, manipulation and transcoding of object content will only be available for NCR TIFF or MO:DCA objects through the `<GetImagePage>` element. Note that a `<GetImagePage>` request with no manipulation or transcoding requested will return the extracted CCITT G3/G4, JPEG or ABIC image.

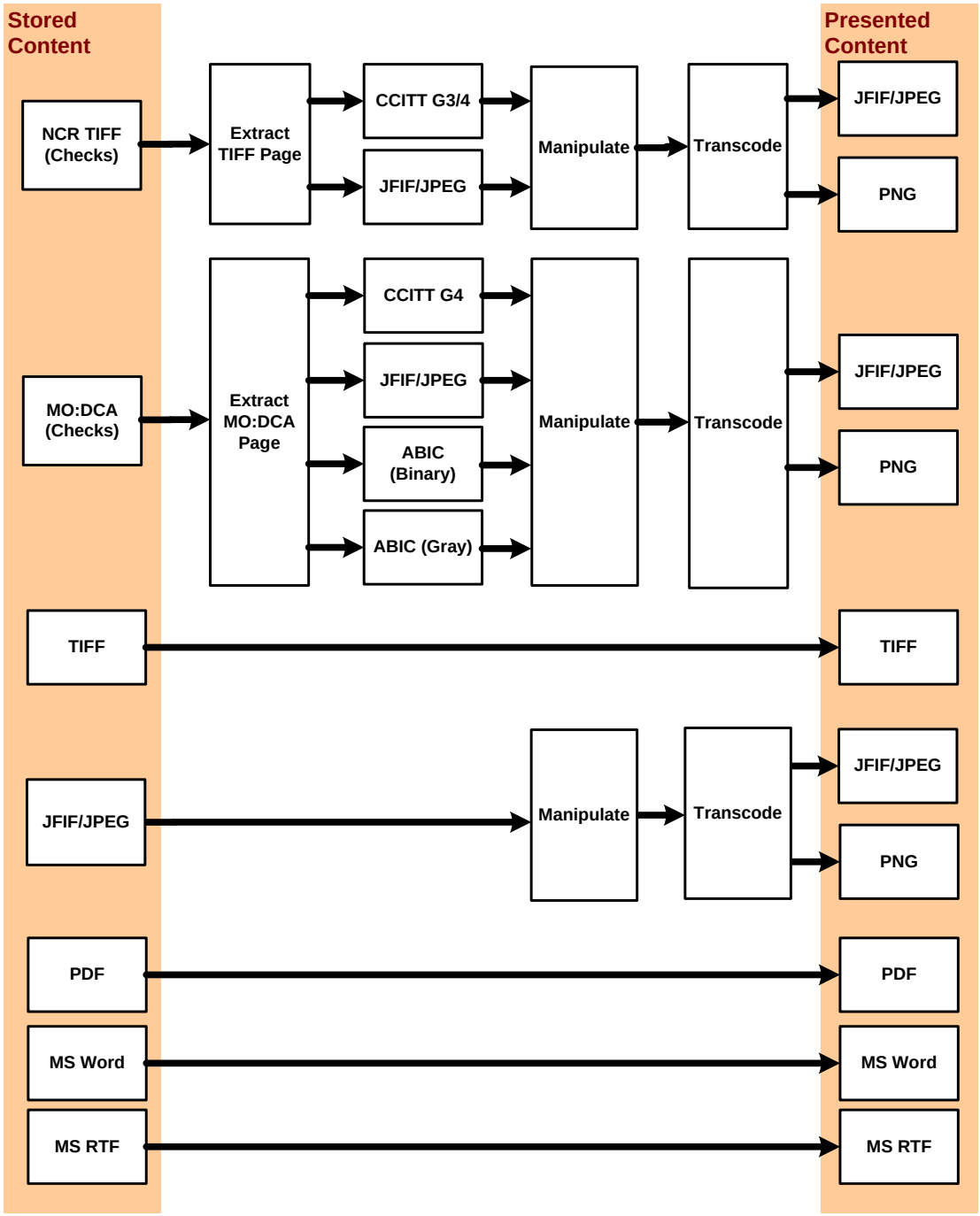


Figure 8-1: Object Content Operations

Table 8-1 below shows the permitted manipulation and transcoding between stored/extracted content types and presented content types.

Presented Content Type	Stored/Extracted Content Type									
	NCRT		MDCA				DOC	PDF	RTF	TIF
	CC3/CC4	JPG	CC4	JPG	ABCB	ABCG				
JPG	X	X	X	X	X	X				
DOC							X			
PDF								X		
PNG	X	X	X	X	X	X				
RTF									X	
TIF										X

Table 8-1: Permitted Content Transcoding and Manipulation

9. Appendix F: Glossary

Term	Meaning
ABIC	**A**daptive **Bi**-level **I**mage **C**ompression. A proprietary IBM image compression algorithm. Despite its name, ABIC now supports both binary and greyscale compression.
ACS	**A**rchive **C**ontent **S**ervices. An ImageMark Archive service that lets requesters retrieve objects through **AIS** and have specified operations performed on the content of these objects, including image manipulations, transcoding and extraction of pages from TIFF or MO:DCA objects. ACS requests and responses are XML documents transmitted using the HTTP protocol.
AFP	**A**dvanced **F**eature **P**resentation (originally Advanced Function Printing). A published IBM standard for data stream architecture. Now generally used as a synonym for Mixed Object Document Content Architecture (see **MO:DCA**), although the AFP definition also includes other, non-MO:DCA content types.
AIS	**A**rchive **I**nteraction **S**ervices. The common interface to ImageMark Archive that lets requesters perform security, retrieval and update operations. AIS requests and responses are XML documents transmitted over direct TCP/IP socket connections.
Base64	A mechanism for encoding arbitrary binary information for transmission. The encoding maps the binary information into a restricted set of the printable ASCII character set. This allows it to be used for imbedding binary information in XML documents without introducing any of the XML reserved characters and making the document invalid.
CC3	The ImageMark Archive CCITT Group 3 content type. CCITT Group 3 is a compression algorithm used for compressing binary image data. Originally developed for fax transmissions but now widely used for all kinds of image data. Binary images in TIFF wrappers are often compressed using the CCITT algorithms. CCITT Group 3 exists in 1-dimensional and 2-dimensional variants.
CC4	The ImageMark Archive CCITT Group 4 content type. CCITT Group 4 is a successor to Group 3 and offers improvements in resiliency.
HTTP	**H**yper**t**ext **T**ransfer **P**rotocol. An application-level protocol for distributed, collaborative, hypermedia information systems. HTTP is the protocol used by Web services and applications. It is the protocol that ACS uses for transmission of its XML request and response documents.
IFD (TIFF)	**I**mage **F**ile **D**irectory. A TIFF header entry that contains information about a single image that is held in the TIFF object. Often referred to as a TIFF *page* and used to describe an image in a multi-image TIFF, rather than just its header entry.
IOCA	**I**mage **O**bject **C**ontent **A**rchitecture. A published IBM standard for an image container used to interchange and present images. Similar in purpose to a TIFF wrapper, an IOCA object can hold one or more image objects. Common image compression formats supported include ABIC, CCITT Group 3, CCITT Group 4, and JPEG.
JPG	The ImageMark Archive JFIF/JPEG content type. JPEG is a compression algorithm used for compressing grey and colour images. JFIF is a very thin wrapper around JPEG images that holds information needed for file interchange purposes.
MO:DCA	**M**ixed **O**bject **D**ocument **C**ontent **A**rchitecture. A published IBM standard for a container object that serves as a wrapper for other objects of various types including: IOCA: Image Object Content Architecture PTOCA: Presentation Text Object Content Architecture GOCA: Graphics Object Content Architecture FOCA Font Object Content Architecture BCOCA: Bar Code Object Content Architecture
NCR TIFF	A subset of the full TIFF specification used by NCR to hold check images and images of other check-size documents. A full specification of the expected TIFF tags is given in reference [2], *ImageMark Capture: Capture System to ImageMark Archive Interface Specification*, but the main restriction is that images be CCITT Group 3, CCITT Group 4 or JPEG compression.

Term	Meaning
Photometrics	Whether a zero-valued pixel is to be shown as black and a maximum-valued pixel is to be shown as white or vice-versa.
PNG	The ImageMark Archive PNG content type. PNG ("Portable Network Graphics") is the image compression standard recommended by W3 for grey and colour images that are to be displayed on Web browsers.
Raster Scan Content	Image content that comes from scanning documents and is represented by an array of pixels or some compressed format that results from transforming this array of pixels. Other types of content may be a *Page Definition Language* (such as PDF or Word) which describes the content in enough detail to allow it to be generated for display but does not actually hold the array of pixels.
TIFF	**T**agged **I**mage **F**ile **F**ormat. A de facto standard used to provide a wrapper for image data. A TIFF object may contain one or more images, with each image being described and located by an IFD entry in the TIFF header. ImageMark Archive uses TIFF objects to hold check images.
Transcoding	The conversion of one content type into another.

10. Appendix G: Content Descriptor Codes

The following table shows codes that are to be used as content descriptors. **Only the codes shown in bold typeface are used in the initial release of Archive Content Services.** The other codes are for future use only. Note that the MIME type shown here is not the MIME type in the HTTP response header. This is always `text/xml containing a base64-encoded image`.

Value	MIME Type	Description
NONE	***none***	**Unknown content type**
ABCB	***none***	IBM ABIC (Binary)
ABCG	***none***	IBM ABIC (Gray)
AFP	`application/vnd.ibm.modcap`	IBM Advanced Feature Presentation. Here used as a synonym for MDCA (MO:DCA)
BMP	`image/x-bmp`	Microsoft Bitmap
CC3	`image/g3fax`	CCITT Group 3 1-D and 2-D
CC4	*none*	CCITT Group 4
DOC	`application/msword`	Microsoft Word
DVI	`application/x-dvi`	LaTex document
GIF	`image/gif`	GIF image file
GZ	`application/x-gzip`	GZIP compressed file
HTM	`text/html`	HTML document
ICA	***none***	IBM IOCA
ICO	`image/vnd.microsoft.icon`	Microsoft Icon
JPG	**`image/jpeg`**	**JFIF/JPEG**
MAN	`application/x-troff-man`	Unix manual page
MDCA	`application/vnd.ibm.modcap`	IBM Mixed Object Document Content Architecture (MO:DCA). Here used as a synonym for AFP
MID	`audio/midi`	MIDI audio
MOV	`video/quicktime`	Apple Quicktime video
MP2	`audio/mpeg2`	Audio MPEG2
MP3	`audio/mpeg3`	Audio MPEG3
MPG	`video/mpeg`	MPEG video
NCRT	`image/ncr-imagemark`	NCR TIFF – a check sized TIFF document with CC3/CC4/JPG front and back images.
PBM	`image/x-portable-bitmap`	Portable Bit Map
PCL	`application/vnd.hp-pcl`	HP PCL printstream
PCX	`application/pcx`	PCX format
PDF	`application/pdf`	Adobe Portable Document Format
PGM	`image/x-portable-graymap`	Portable Gray Map

Value	MIME Type	Description
PNG	**`image/png`**	**Portable Network Graphics**
PNM	*none*	Portable Any Map
PPM	`image/x-portable-pixmap`	Portable Pixel Map
PPT	`application/vnd.ms-powerpoint`	Microsoft Powerpoint
PS	`application/postscript`	Adobe Postscript
RTF	`application/rtf`	Rich Text Format
TAR	`application/x-tar`	Composite TAR file
TIF	`image/tiff`	TIFF - Tagged Image File Format. We use this for non-check TIFFs. Check TIFFs are type "NCRT"
TXT	`text/plain`	Plain text format
WAV	`audio/wav`	Microsoft WAV file
WP	`application/wordperfect`	Corel WordPerfect
WRL	`model/vrml`	VRML model
XLS	`application/vnd.ms-excel`	Microsoft Excel
XML	`text/xml`	XML document
ZIP	`application/zip`	ZIP compressed file

11. Appendix H: MO:DCA Object Formats

The MO:DCA object format, as defined in reference [13], *Mixed Object Document Content Architecture (MO:DCA) Reference* , is a complex and flexible data structure that can hold image, text, graphic and bar code objects, as well as resources that define how these are to be presented. However, ACS is only concerned with a small subset of this structure and can ignore most of this complexity. *Figure 11-1* below shows the components of the MO:DCA structure that concern ACS, and the following paragraphs describe important points. There are other components which are not needed by ACS and may be ignored.

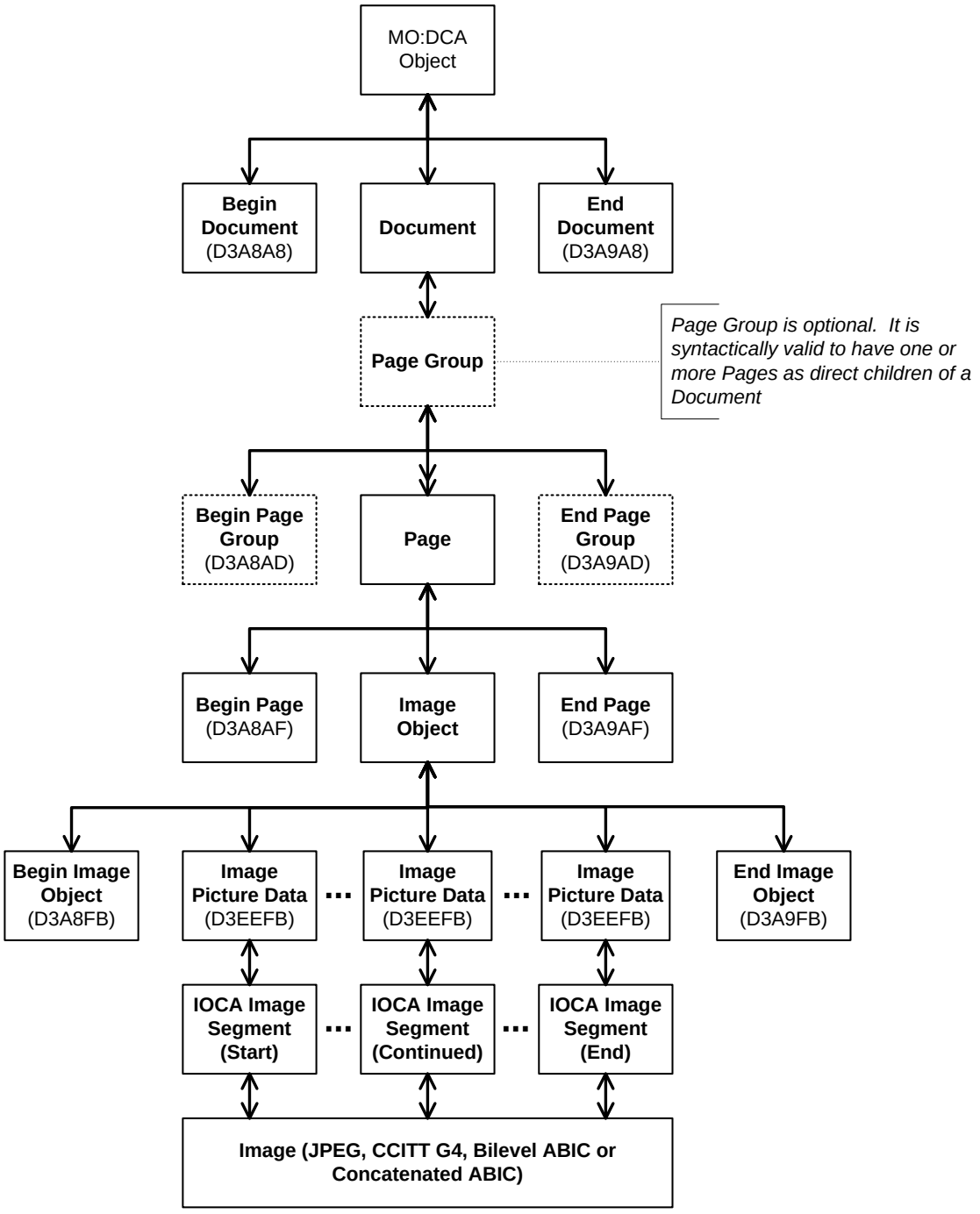


Figure 11-1: MO:DCA Object Structure

The following points must be considered when processing a retrieved MO:DCA object:

- Each MO:DCA object holds a single document. This may be a check item, a bank statement or some other type of document. Often, MO:DCA objects are used as print streams, where they may hold many documents. In the case of ImageMark Archive, this will not be the case and a single MO:DCA object will only hold one document.
- The MO:DCA object will hold a single, top-level Document component. The Document component will start with a Begin Document tag, will either hold a single Page Group child component or one or more Page components. and will end with an End Document tag.
- If there is a Page Group component this will start with a Begin Page Group component, followed by one or more Page components and end with an End Page Group component.
- Each Page component will start with a Begin Page component, will contain a single Image Object, and will end with an End Page component. It can be assumed that the order of Page components within the parent Document or Page Group is the order of pages within the document, so the first Page component encountered will hold the image of the first page of the document.
- Each Image Object will hold a single image, which will be of one page of the document. The Image Object component starts with a Begin Image Object tag, followed by one or more Image Picture Data components, and ending with an End Image Object tag.
- Each Image Picture Data component either contains a complete IOCA Image Segment or part of an IOCA Image Segment. Where there is more than one Image Picture Data component in an Image Object, the partial IOCA Image Segments extracted from each Image Picture Data component, must be concatenated to make the complete IOCA Image Segment.
- Each complete IOCA Image Segment holds an image in one of the following formats:
  - CCITT Group 4 binary
  - JPEG greyscale
  - Bilevel ABIC binary
  - Concatenated ABIC greyscale

12. Appendix I: Archive Content Services Image Retrieval

All requests to ACS must be made using the **HTTP POST** method. All conversations between an ACS client and ACS are made by exchanging XML messages. The request must be sent using HTTP POST to the ACS server and the XML containing the request must be part of the HTTP POST payload. ACS then processes the request and returns an XML message as its response to the ACS client. The URL to post the request is in the following format (optional elements are in [] brackets)

http[s]://<ACS Server host>[:<ACS Server port>]**/acs/servlet/acs**

- http OR https – Use http for non SSL requests and https for SSL requests
- <ACS Server host> – The hostname of the server where ACS is hosted e.g. localhost, www.acsserver.com, 123.456.789.12 (IP addresses), etc…
- <ACS Server port> – If ACS is hosted on a non-standard port (other than 90 for http OR 443 for https) we need to give the port number in the request URL
- /acs/servlet/acs – This part remains the same for ALL requests irrespective of hostname, SSL / non SSL, default or non-default port numbers

Sample ACS request URL's are as follows:

- https://www.acsserver.com/**acs/servlet/acs**
- https://123.456.789.123/**acs/servlet/acs**
- https://123.456.789.123:9080/**acs/servlet/acs**

12.1 Retrieving Images using ACS

To retrieve an image from ACS, do the following steps:

1. Send a login request XML and retrieve a "sectoken" from the ACS response.
2. Send an image retrieval request XML passing the sectoken retrieved in step 1.
3. Send a logout request XML to invalidate the "sectoken" when we are done with retrieving images.

Note: We can issue multiple image retrieval requests using the same sectoken until the sectoken expires OR until a logout request is sent. When the sectoken expires while retrieving the image instead of an image an error is returned back from ACS indicating that the sectoken has expired. See error codes section for the exact error codes indicating that the sectoken is no longer valid.
As a best practice we should send a logout request once we are done retrieving images. This can be done after each image retrieval request however for better performance it is recommended to logout after the user logs out from the host application. This prevents the additional overhead incurred in ACS during each login request which will improve overall speed of image retrieval.
If the sectoken expires we need to send a new login request, get a new sectoken and use the new sectoken for retrieving images.

12.2 Sample Request / Response for Retrieving Images from ACS

12.2.1 Login to get a sectoken

A sample login request is as follows:

```
<?xml version="1.0"?>
<loginmanager sessionname="session" applname="Cheque-inquiry">
<login user="John" domain="system" pwd_encoding="MD5" password="54b53072540eeeb8f8e9343e71f28176"
autologoff="0"/>
</loginmanager>
```

A sample response for the above request is as follows:

```
<?xml version=”1.0” ?>
<loginmanagerresult sessionname=“session” code=”0” subcode=”0”>
<loginresult code=“0” subcode=”30” sectoken=”4db3afda35906fe65b5cda7dbbabb68a00000000” />
</loginmanagerresult>
```

Note:

- The username / password / applname are setup by the system administrator. Please reach out to them for details of the same.
- The sessionname attribute is simply echoed back on each request and can be used for debugging purposes.
- The sectoken highlighted in bold above will be unique for each login request and must be sent on all subsequent requests until we send a logout request OR the sectoken expires.

12.2.2 Image retrieval request

A sample image retrieval request is as follows. The sectoken retrieved from the login request earlier needs to be sent again as part of the image retrieval request.

```
<?xml version="1.0"?>
<inquiry sectoken="4db3afda35906fe65b5cda7dbbabb68a00000000" applname='Cheque-inquiry' sessionname='session' priority="1"
images="Y" mediastatus="Y">
        <object name = "cheques" />
        <resultfield name="ACCOUNT"/>
        <resultfield name="AMOUNT"/>
        <resultfield name="PAYORBANKNUMBER"/>
        <indexquery queryname="Online Query">
                <qfield name="UDK" low="1" high="1"/>
                <qfield name="ClearingHouseIdentifier" low="1" high="1"/>
                <qfield name="CheckNumber" low="1667319" high="1667319"/>
                <qfield name="PayorBankNumber" low="600229002" high="600229002"/>
                <ProcessContent>
                        <GetTiffPage Pages="1-3">
                                <TranscodeContent RequestedContentDescriptor="PNG" />
                        </GetTiffPage>
                </ProcessContent>
        </indexquery>
</inquiry>
```

NOTE: The above request is a sample request and the exact values for applname, resultfield name / qfield name etc… are determined by the archive object type definition. Please contact the system administrator responsible for setting up the archive object definition for necessary details required to make the image retrieval request.

A sample response for the above request is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<inquiryresult sessionname="session" code="0" subcode="0">
<indexqueryresult code="0" subcode="0" queryname="Online Query">
<item objectname="testobject">
```

<rfield name="ACCOUNT" value="17424"></rfield>
<rfield name="AMOUNT" value="2002"></rfield>
<rfield name="PAYORBANKNAME" value="1"></rfield>
<ProcessContentResult code="0" subcode="0">
<GetImagePageResult PageNumber="1" code="0" subcode="0">
<image status="0" original_len="16950" encoded_len="22913"
encode_type="base64">iVBORw0KGgoAAAANSUhEUgAAAiYAAASoAQAAAAAI3zx5AAAgAElEQVR4nO3dy68jV3oY8jV3oY8KJp
...
...
...
...
...
</image>
</GetImagePageResult>
</ProcessContentResult>
</item>
</indexqueryresult>
</inquiryresult>

NOTE: The image retrieved in the <image>…</image> tag is Base 64 encoded and must be decoded to get the original image in binary form

12.2.3 Logout to invalidate sectoken

A sample logout request is as follows:
<?xml version="1.0" ?>
<loginmanager sessionname="session" applname="Cheque-inquiry" sectoken="4db3afda35906fe65b5cda7dbbabb68a00000000">
<logout />
</loginmanager>

A sample response for the above request is as follows:

<?xml version=”1.0”?>
<loginmanagerresult sessionname=”session” code=“0” subcode=”1234567”>
<logoutresult code=“0” subcode=”0”/>
</loginmanagerresult>

The logoutresult code=0 and subcode=0 indicate that the logout was successful

NOTE: Once a logout request is made the sectoken will become invalid and we must send a new login request to retrieve another sectoken to continue retrieving images.

13. Appendix J: Frequently Asked Questions (FAQs)

13.1 Multiple login instances using a single login account to ACS

This is a scenario where a single account has been setup for ACS and the bank has multiple users:
After a login request for the first user is made, for subsequent login requests ACS responds back with error code 21 and error sub code 4 for the second, third, fourth users… and so on.

Following is the Login Response XML for the second, third user etc… even though first user login request returned a valid sectoken.

```
<?xml version="1.0" encoding="UTF-8"?><loginmanagerresult sessionname="Session6822" code="0" subcode="0">
   <loginresult code="21" subcode="4"></loginresult>
</loginmanagerresult>
```

This response indicates that the ACS user has already been given a sectoken and by default only one ACS user can receive a sectoken at a given time. In this case we must enable allow multiple logins for the account setup for ACS using WebSAT. This setup would be done by the system administrator on their network.

13.2 Validity of a sectoken and handling sectoken expiry

The duration that a sectoken remains valid is dependent on the configurations made in AIS by the by the system administrator on their network. It is not possible to know before making the request if the current session associated with a sectoken is active or not. Hence following is the recommended logic while working with ACS:

- When a login request is made the sectoken must be stored in the session level scope so that it can reused across multiple requests for a given user.
- Storing the sectoken in the session level scope would require that a new sectoken be requested through a login request to ACS for each user and each user would have their own separate sectoken stored in their session.
- After a valid sectoken has been obtained at any given time if ACS responds back indicating that the session has timed out / expired a new login request must be sent automatically to get a new sectoken followed by resending the original request with the new sectoken. The session level scope must be updated with the new sectoken so that subsequent requests for the user happen using the new sectoken.
- When the user logs out of the host application or when the session scope on the host application is about to expire for the user, a logout request must be sent automatically for that sectoken to invalidate it.
- From a security standpoint we strongly recommend that the sectoken be stored on the server side in the session scope rather than in a form variable etc… in the HTML page sent to the user. The sectoken should never be transmitted in any way to the user, it should only be sent over the network while communicating with ACS.

13.3 Session name usage

Please note that the session name is meant for debugging purposes. ACS will echo back the session name passed in the request and it can be used for tracking individual requests for debugging.

13.4 AIS services valid through ACS

All requests to ACS must be made using the AIS online queries. Batch AIS queries are not supported through ACS and should not be used.

13.5 Usage of image manipulations and format of images retrieved through ACS

The image manipulation operations such as scale, rotate etc… within the ManipulateRasterContent tag are a feature of ACS however they are optional and the bank may choose to not use them, using ACS only for image retrieval in that case. Similarly, usage of either JPG or PNG etc… for the format of the image (Content Descriptor Codes) retrieved is as per the banks discretion.