How to Set up and Use the NCIP Server

Version 18.01 and later

CONFIDENTIAL INFORMATION

DISCLAIMER

TRADEMARKS

Document released: August 31, 2014

Web address: http://www.exlibrisgroup.com

Table of Contents

1 General

The NISO Circulation Interchange Protocol (NCIP) defines a repertoire of messages and associated rules of syntax and semantics for use by applications to:

- Perform the functions necessary to lend items.
- Provide controlled access to electronic resources.
- Facilitate co-operative management of these functions.

The standard specifically addresses conditions in which the applications that initiate the lending of items or control of access must transmit information about the user, agency, items, and/or access that is essential to successful conclusion of the function.

NCIP is a connection-oriented, sessionless, protocol.

- Connection-oriented - Circulation processes happen in real time, often with the user present and awaiting service. A connection-oriented protocol facilitates a timely interaction between applications and allows the application requesting a service to know with confidence that a message was received by the partner application.
- Sessionless - The lifecycle of a particular circulation activity provided by an agency to a user is often extended over days, weeks, or months. It is therefore impractical to maintain sessions between two applications.

An NCIP message must be valid XML that conforms to the NCIP DTD (Document Type Definition). The NCIP DTD defines the content of an NCIP message as a series of elements, each of a complex, simple or EMPTY type.

2 Transport Layer

The NCIP server is supported either over a TCP/IP transport layer or over HTTPS.

When TCP/IP is used, a special NCIP server port is used as the access point to the NCIP server. In that case the NCIP server activity is logged in a separate log file. Many separate NCIP servers can be run, each with its own configuration in the tab_ncip.conf table, and each serving a single ADM library. More than one NCIP server may serve the same ADM library.

When HTTPS is used, the access to the NCIP server is done through ALEPH's web port. Each initiator is assigned an ADM library to which its NCIP messages will be directed. The way this is done is described below in the Setup chapter. The NCIP server activity will be logged in the www_server log file.

3 LookupVersion

The LookupVersion message is supported and can be used to report the NCIP protocol version number that is supported. The structure of the supported message and response is defined in the DTD file http://www.niso.org/ncip/v1_0/imp1/dtd/ncip_version.dtd

Successful Result: The responding application returns the requested data.

4 Supported Services

Out of the 45 messages that are defined in the NCIP Protocol, 7 are supported by the ALEPH NCIP server. The messages are supported according to the DTD http://www.niso.org/ncip/v1_0/imp1/dtd/ncip_v1_0.dtd.

4.1 Authenticate User Service

This service requests authentication of a User presumed to be known to an Agency. Authentication indicates only that the User is known by the responding Agency. The initiating application must determine the type of data the responding application requires for authentication.

Successful Result: The responding application authenticates the User and returns the unique ID of the User.

4.2 Lookup Item Service

This service requests data about a particular Item known to the responding application. The initiator provides the unique ID of the Item and a list of elements for which data is requested.

Successful Result: The responding application returns the requested data to the initiating application.

4.3 Lookup User Service

This service requests data about a particular User known to the responding application. The initiator provides the unique ID of the User and a list of elements for which data is requested.

Successful Result: The responding application returns the requested data to the initiating application.

4.4 Check Out Item Service

This service requests that the responding application check out an Item to a User. The initiating application may also request data about the User and/or Item involved with this check-out.

Successful Result: The responding application checks out the Item to the User until the date indicated in the response. It may also supply the data elements requested.

Note
The NCIP Server consults the "tab_attr_sub_library" settings (in the ADM library).
A "check out" service is only performed if the IP address of the sending application is registered as type 1 in Column 2 of tab_attr_sub_library.

4.5 Check In Item Service

This service requests that the responding application check in an Item. It also permits the initiating application to request data about the User and/or Item involved with this check in.

Successful Result: The responder checks in the Item and returns requested User or Item data.

Note
The NCIP Server consults the "tab_attr_sub_library" settings (in the ADM library).
A "check in" service is only performed if the IP address of the sending application is registered as a type 1 or 2 in Column 2 of tab_attr_sub_library.

4.6 Request Item Service

This service requests that the responding application place a request on an Item for a User whether or not the Item is immediately available. The initiating application indicates the type of request being made. The initiating application may also request data about the User and/or Item involved with this request.

Successful Result: The responding application places the request and provides data about where the Item may be picked up and the date it expects the Item to be available. It may also supply the data elements requested.

Note
The request will be placed with the pickup location set to the value that is specified as the FromAgencyId. Please refer to the chapter that describes Setting up Agencies for the required configurations.

4.7 Cancel Request Item Service

This service requests that the responding application cancel a previous request for an Item. The initiating application may also request data about the User and/or Item involved with this request cancellation.

Successful Result: The responding application cancels the request. It may also supply the data elements requested.

4.8 Accept Item Service

This service requests that the responding application accept an Item to be circulated to a User. The responding application may be a third party that has no prior knowledge of either the User or the Item. If there is a possibility that the responding application has no prior knowledge of either the User or the Item, the request may optionally include data about the User and/or the Item.

Successful Result: The responder accepts the Item for the action specified. It returns the Unique Request ID. It will also provide a Unique Item ID that the initiating application may use to reference the Item in subsequent messages.

Detailed information on the actions taken by the ALEPH system upon receiving an AcceptItem message can be found in a separate chapter (See chapter 6).

5 Supported Element and Values

The messages and responses that are supported by the ALEPH NCIP server conform to the DTD http://www.niso.org/ncip/v1_0/imp1/dtd/ncip_v1_0.dtd. The elements and values that are described below are taken from the above mentioned DTD.

5.1 General Elements

This chapter describes the supported elements that are common to many of the supported services.

5.1.1 The Response Header

The Response Header element in all of the NCIP server's responses uses the following format:

ELEMENT ResponseHeader (FromAgencyId , ToAgencyId)

The values of the FromAgencyId elements will be those that were used in the initiating message as ToAgencyId.

The values of the ToAgencyId elements will be those that were used in the initiating message as FromAgencyId.

5.1.2 The Item Element Type and Item Optional Fields

The following values of the ItemElementType element are supported:

Item Description

BibliographicDescription

Circulation Status

If the ItemElementType element is used in the message with the value 'Item Description', the ItemOptionalFields element will be found in the response.

The following structure of the response ItemOptionalFields element is supported.

ItemOptionalFields

BibliographicDescription

Author

BibliographicItemId

BibliographicItemIdentifier

BibliographicRecordIdentifier

PublicationDate

Publisher

Title

MediumType

ItemDescription

VisibleItemId

CallNumber

Circulation Status

Either *BibliographicItemIdentifier* or *BibliographicRecordIdentifier* will be returned, depending on:

If the incoming message uses one of these elements, the response will include the same element.

If the incoming message did not use these elements then if the bibliographic_item_id variable of tab_ncip.conf is ISSN or ISBN, then *BibliographicItemIdentifier* will be returned. In any other case *BibliographicRecordIdentifier* will be returned.

5.1.3 The User Element Type and User Optional Fields

The following values of the UserElementType are supported:

Name Information

User Address Information

User Privilege

Visible User Id

If the UserElementType element is used in the message with the value 'Name Information', the UserOptionalFields element will be found in the response with the NameInformation element.

If the UserElementType element is used in the message with the value 'User Address Information', the UserOptionalFields element will be found in the response with the UserAddressInformation element

If the UserElementType element is used in the message with the value 'User Privilege', the UserOptionalFields element will be found in the response with the UserPrivilege element.

If the UserElementType element is used in the message with the value 'Visible User Id', the UserOptionalFields element will be found in the response with the UserVisibleId element.

The following structure of the response UserOptionalFields element is supported.

UserOptionalFields

NameInformation

PersonalNameInformation

UnstructuredPersonalUserName

VisibleUserId

UniqueAgencyId

VisibleUserIdentifierType

VisibleUserIdentifier

UserAddressInformation

UserAddressRoleType

ValidFromDate

ValidToDate

PhysicalAddress

UnstructuredAddress

UnstructuredAddressType

UnstructuredAddressData

PhysicalAddressType

ElectronicAddress

ElectronicAddressType

ElectronicAddressData

UserPrivilege

UniqueAgencyId

AgencyUserPrivilegeType

ValidFromDate

ValidToDate

UserPrivilegeDescription

Note that the UserAddressInformation element will contain either PhysicalAddress OR ElectronicAddress. If the user has both, then the returned UserOptionalFields will have two UserAddressInformation elements, one with the PhysicalAddress element and one with the ElectronicAddress element.

5.2 Services

This chapter describes the elements that are supported in the message and in the response of each of the supported services.

5.2.1 Authenticate User Service

5.2.1.1 Message

As all elements of the message are mandatory, they are all supported. The following structure is therefore supported:

AuthenticateUser

InitiationHeader

AuthenticationInput

AuthenticationInputData

AuthenticationDataFormatType

AuthenticationInputType (Barcode Id,Password)

5.2.1.2 Response

The user is authenticated by using the first value of the AuthenticationInputData element as key data, and the second as key verification.

As all elements of the response are mandatory, they are all supported. The following structure is therefore supported:

AuthenticateUserResponse

ResponseHeader

Problem

OR

UniqueUserId

5.2.2 Lookup Item Service

5.2.2.1 Message

The following structure of the LookupItem message is supported:

LookupItem

InitiationHeader

UniqueItemId OR VisibleItemId (Barcode)

ItemElementType

5.2.2.2 Response

The following structure of the LookupItem response is supported:

LookupItemResponse

ResponseHeader

Problem

OR

UniqueItemId

ItemOptionalFields

5.2.3 LookupUser Service

5.2.3.1 Message

The following structure of the LookupUser message is supported:

LookupUser

InitiationHeader

UniqueUserId

OR

VisibleUserId

OR

AuthenticationInput (See 5.2.1.1)

UserElementType

5.2.3.2 Response

The following structure of the LookupUser response is supported:

LookupUserResponse

ResponseHeader

Problem

OR

UniqueUserId

UserOptionalFields

5.2.4 CheckOutItem Service

5.2.4.1 Message

The following structure of the CheckOutItem message is supported:

CheckOutItem

InitiationHeader

UniqueUserId

OR

AuthenticationInput (See 5.2.1.1)

UniqueItemId

ItemElementType

UserElementType

5.2.4.2 Response

The following structure of the CheckOutItem response is supported:

CheckOutItemResponse

ResponseHeader

Problem

OR

UniqueItemId

UniqueUserId

DateDue

ItemOptionalFields

UserOptionalFields

5.2.5 CheckInItem Service

5.2.5.1 Message

The following structure of the CheckInItem message is supported:

CheckInItem

InitiationHeader

UniqueItemId

ItemElementType

UserElementType

5.2.5.2 Response

The following structure of the CheckInItem response is supported:

CheckInItemResponse

ResponseHeader

Problem

OR

UniqueItemId
UniqueUserId
RoutingInformation
RoutingInstructions
Destination
Location
LocationType
LocationName
LocationNameInstance (Up to 3 values – sub library, collection and call number)
LocationNameLevel
LocationNameValue
ItemOptionalFields
UserOptionalFields

5.2.6 Request Item Service

5.2.6.1 Message

The following structure of the Request Item message is supported:

RequestItem

InitiationHeader

UniqueUserId
OR
AuthenticationInput (See 5.2.1.1)

UniqueItemId
OR

UniqueBibliographicId

BibliographicItemId

BibliographicItemIdentifier

BibliographicItemIdentifierCode (ISBN or ISSN)

OR

BibliographicRecordId

BibliographicRecordIdentifier

BibliographicRecordIdentifierCode

RequestType (values can be 'Hold' or 'Stack Retrieval')

RequestScopeType

NeedBeforeDate

ItemElementType

UserElementType

ShippingNote (v.20 and later)

The use of the BibliographicItemId and BibliographicRecordId elements depends on the configuration of the tab_ncip_record_id table. Please see part 7.1.2.

5.2.6.2 Response

The following structure of the Request Item response is supported:

RequestItemResponse

ResponseHeader

Problem

OR

UniqueItemId

UniqueRequestId

UniqueUserId

RequestType

(values can be 'Hold' or 'Stack Retrieval')

RequestScopeType

ItemOptionalFields

UserOptionalFields

5.2.7 Cancel Request Item Service

5.2.7.1 Message

The following structure of the Cancel Request Item message is supported:

CancelRequestItem

InitiationHeader

UniqueUserId

OR

AuthenticationInput (See 5.2.1.1)

UniqueItemId

OR

UniqueRequestId

RequestType (values can be 'Hold' or 'Stack Retrieval')

RequestScopeType

ItemElementType

UserElementType

5.2.7.2 Response

The following structure of the Cancel Request Item response is supported:

CancelRequestItemResponse

ResponseHeader

Problem

OR

UniqueItemId

UniqueRequestId

UniqueUserId

ItemOptionalFields

UserOptionalFields

Note that if the UniqueItemId is supplied it will be the only used value to identify the request.

NOTE : Up to version 16.02, in order to use CancelRequestItem with the UniqueRequestId element it is required to define a direct index in the ADM library on the RID field.

5.2.8 Accept Item Service

5.2.8.1 Message

The following structure of the Accept Item message is supported:

AcceptItem

InitiationHeader

UniqueRequestId

RequestedActionType

UniqueUserId

UniqueItemId

ItemOptionalFields

BibliographicDescription

5.2.8.2 Response

The following structure of the AcceptItem response is supported:

AcceptItemResponse

ResponseHeader

Problem

OR

UniqueRequestId
UniqueItemId

6 ALEPH as a Pickup Agency

Upon receiving an AcceptItem message, ALEPH will take the following actions:

- Create a temporary BIB record in the ILL library.
- Create a temporary ADM record attached to the BIB record.

- Create a temporary item that is attached to the ADM record.
- Create a hold request on the temporary item.

These actions closely resemble how ALEPH handles receiving items that are required for fulfilling an ILL request.

Managing the BIB record and the item is done in the same manner that the temporary BIB records created for ILL requests are handled.

Note that for this purpose you must have an ILL library (xxx20 where the old ILL is used or xxx40 where the Integrated ILL module is used) in your system. It is recommended to create the library based on the Aleph ILL library template. The library must have an ILL relation to the ADM library in which the NCIP server is run. For example, put the following line in `$alephe_tab/library_relation`:

- Where the Old ILL module is used:

  ```
  ILL USM50 USM20
  ```

- Where the Integrated ILL module is used:

  ```
  ILL USM40 USM50
  ```

No other ILL setup is required.

6.1 Building the BIB Record

The BIB record will be created depending on the ItemOptionalFields / BibliographicDescription subelements, according to the following guidelines. All values are mapped into subfield “a” unless otherwise stated:

- If the element includes a ‘Author’ element then its value will be mapped depending on the system’s tab 100 MARC-TYPE variable in the following way:
  - If the MARC type is 2 (UNIMARC) then the author information will be mapped to field "700 1".

  In any other case it will be mapped to “100”.
- If the element includes an ‘AuthorOfComponent’ element then its value will be mapped depending on the system’s tab 100 MARC-TYPE variable in the following way:
  - If the MARC type is 4 (MAB) then the title information will be mapped into field "104”.
  - If the MARC type is 2 (UNIMARC) then the title information will be mapped into field "701".

  In any other case the title information will be mapped into field "7001".
- If the element includes a ‘Title’ element then its value will be mapped depending on the system’s tab 100 MARC-TYPE variable in the following way:

- If the MARC type is 4 (MAB) then the title information will be mapped into field "331".
- If the MARC type is 2 (UNIMARC) then the title information will be mapped into field "2001".

In any other case the title information will be mapped into field "2451".

- If the element includes a 'TitleOfComponent' element then its value will be mapped depending on the system's tab 100 MARC-TYPE variable in the following way:
  - If the MARC type is 4 (MAB) then the title information will be mapped into field "ART".
  - If the MARC type is 2 (UNIMARC) then the title information will be mapped into field "517".

  In any other case the title information will be mapped into field "74002".
- If the element includes an 'Edition' element then its value will be mapped depending on the system's tab 100 MARC-TYPE variable in the following way:
  - If the MARC type is 4 (MAB) then the title information will be mapped into field "403".
  - If the MARC type is 2 (UNIMARC) then the title information will be mapped into field "205".

  In any other case the title information will be mapped into field "250".
- If the element includes a 'PublicationDate' element then its value will be mapped depending on the system's tab 100 MARC-TYPE variable in the following way:
  - If the MARC type is 4 (MAB) then the title information will be mapped into field "425".
  - If the MARC type is 2 (UNIMARC) then the title information will be mapped into field "210" and in subfield "d".

  In any other case the title information will be mapped into field "260" and in subfield "c".
- If the element includes a 'Publisher' element then its value will be mapped depending on the system's tab 100 MARC-TYPE variable in the following way:
  - If the MARC type is 4 (MAB) then the title information will be mapped into field "412".
  - If the MARC type is 2 (UNIMARC) then the title information will be mapped into field "210" and in subfield "c".

  In any other case the title information will be mapped into field "260" and in subfield "b".

- If the element includes a ‘PlaceOfPublication’ element then its value will be mapped depending on the system’s tab 100 MARC-TYPE variable in the following way:
    - If the MARC type is 4 (MAB) then the title information will be mapped into field "410”.
    - If the MARC type is 2 (UNIMARC) then the title information will be mapped into field "210" and in subfield “a”.

  In any other case the title information will be mapped into field "260" and in subfield “a”.
- If the element includes a BibliographicItemIdentifierCode/Value element of ISSN then the value of the BibliographicItemIdentifier field will be mapped depending on the system’s tab 100 MARC-TYPE variable in the following way:
    - If the MARC type is 4 (MAB) then the title information will be mapped into field "542".
    - If the MARC type is 2 (UNIMARC) then the title information will be mapped into field "011".

  In any other case the title information will be mapped into field "022"
- If the element includes a BibliographicItemIdentifierCode/Value element of ISBN then the value of the BibliographicItemIdentifier field will be mapped depending on the system’s tab 100 MARC-TYPE variable in the following way:
    - If the MARC type is 4 (MAB) then the title information will be mapped into field "540".
    - If the MARC type is 2 (UNIMARC) then the title information will be mapped into field "010".

  In any other case the title information will be mapped into field "020"
- If the element includes a MediumType/Value element then it will be mapped into the TYP field.

6.2 Building the Item Record and the Request

An item will be created with the following values:

- Z30-SUB-LIBRARY – The Z303-HOME-LIBRARY of the patron for whom the item is created. This is the patron whose ID is included in one of the following fields:
    - UserIdentifierValue (first priority for this value)
    - VisibleUserIdentifier (second priority for this value)
    - FromAgencyId/UniqueAgencyId/Value

- Z30-MATERIAL is BOOK if <BibliographicItemIdentifierCode> is ISBN and ISSUE if <BibliographicItemIdentifierCode> is ISSN.
- Z30-ITEM-STATUS will be hardcoded 98.
- Z30-CATALOGER will be NCIP.
- Z30-ITEM-PROCESS-STATUS will be hardcoded IL. This will cause the item to be grabbed when returned with a message indicating that the item needs some special processing.
- Z30-COLLECTION will be hardcoded NCIP.
- Z30-NOTE-INTERNAL will be created with the InitiationHeade/FromAgencyID/UniqueAgencyId/Value field.

- Z30-BARCODE –
  The barcode with which ALEPH will identify the new item is created according to the tab100 and tab_checksum settings, as configured in the ADM library.

- Z30-CALL-NO
  The first Item Call Number will be created from two elements: the <Value> of the <UniqueAgencyId> element and the <ItemIdentifierValue> element.
  For example, the following input in AcceptItem message will create an item with the call number: "FOREIGN ITEM AGENCY-TEST803":

```
<UniqueAgencyId>
  <Scheme>URSA Agency IDs</Scheme>
  <Value>Foreign Item Agency</Value>
</UniqueAgencyId>
<ItemIdentifierValue>test803</ItemIdentifierValue>
```

- Hold Request:
  A hold request will be created. The hold request will be created for the patron that is supplied in the UniqueUserId field, in the same way it is done when the RequestItem messages is accepted. The pickup location of hold request will be the Z303-HOME-LIBRARY of the patron. In version 19.01 and up, when the parameter "get_pickup_location" is set to Y, the pickup location of the hold request will be taken from the "ToAgencyId" tag.

  The default is "N" (hold request will be the Z303-HOME-LIBRARY of the patron).

7 Setup

7.1 ADM Tables

The tables described in this chapter reside in the ADM in which the NCIP server is active. The ADM library of the NCIP server is the connected library when the server was run using util/w/3/7/2.

7.1.1 tab_ncip.conf

This table defines different parameters for the NCIP server. Different configuration tables can be defined for different NCIP servers by using the server number as suffix. For example, tab_ncip.conf.5160 can be used for an NCIP server that is running on port 5160. If no explicit table is defined with the port number as suffix, the unsuffixed tab_ncip.conf table will be used.

The following parameters can be defined in this table:

Under the [NCIPMessage] section and the [NCIPVersionMessage] section:

- version - Defines the NCIP version that is supported by this server.
- doctype - Defines the doctype that will be included in the header of the server's responses.

The [NCIPMessage] section is used by all messages other than the LookupVersionResponse, which uses the [NCIPVersionMessage] section.

Under the [Main] section:

- dtd_name - Defines the DTD that is used by the NCIP server.
- pin_required - A Y/N option that defines whether the LookupUser message will require authentication using the AuthenticationInput element, or whether it can be accepted without authetication, by using elements such as UniqueUserId or VisibleUserId.
- bibliographic_item_id - Defines the bibliographic record access code. This variable is a string, such as:

  ISBN

  ISSN

  OCLC

  When the server responds with an NCIP response that includes a BibliographicDescription element, the value that will be included in the BibliographicRecordIdentifierCode or BibliographicItemIdentifierCode sub-element is the one specified in this variable. The BibliographicRecordIdentifier or BibliographicitemIdentifier field will be filled in with bibliographic information from the field that is assigned in tab_ncip_record_id for the access code that is set up in this field.

For example, if a LookupItem message is sent for an item, the server will check the bibliographic_item_id variable to know what bibliographic information it should return. If it says OCLC, the server will go to the tab_ncip_record_id table to see what bibliographic field should be used to supply OCLC numbers.

Under the [Default] section:

match_id_type - Defines the ID type that is used for patron authentication by the NCIP server. To authenticate either by ID type 00 or ID type 01, define “auto” in this variable.
If you do not set this value and leave it blank, the system automatically matches any ID that is defined in tab_bor_id.lng to permit a GUI logon (column 5 = Y).

- accept_item_status – Defines the item status that will be assigned to the temporary item which is created by the AcceptItem message.
- get_pickup_location – (version 19.01 and up) When set to "Y", it defines the pickup location of the hold request on the temporary item that is created by the AcceptItem message. The pickup location is retrieved from the "ToAgencyId" tag.
  When this parameter is set to "N" (this is the default) the pickup location is created from the Z303-HOME-LIBRARY of the patron for whom the item is created.
- Return_during_loan – (version 21 and up). If this parameter is set to Y, and the system detects that the loaned item is already loaned to another patron, return during loan is automatically performed.

Under the [DEBUG] section:

- verbose_msg and debug_level define the NCIP server log format.

7.1.2 tab_ncip_record_id

This table defines what record keys will be used by the NCIP server for two purposes:

- Searching for records based on bibliographic information.

  If an incoming message, such as RequestItem, uses the UniqueBibliographicId element to request locating an item based on bibliographic information, the method for handling the request is specified in this table. The table specifies for each record id type (specified in the BibliographicItemIdentifierCode\value or BibliographicRecordIdentifierCode\value element of the incoming message) what bibliographic field should be searched.

- Including bibliographic information in NCIP message responses.

  When the NCIP response includes a BibliographicDescription element, the BibliographicRecordIdentifier or BibliographicitemIdentifier field will be filled in with bibliographic information from the field that is assigned in tab_ncip_record_id

for the access code that is setup in the bibliographic_item_id variable of tab_ncip.conf.

For example, if a LookupItem message is sent for an item, the server will look at the bibliographic_item_id variable to know what bibliographic information it should return. If it says OCLC, the server will go to the tab_ncip_record_id table to see what bibliographic field should be used to supply OCLC numbers.

7.1.3 DTD

The Document Type Definition (DTD) defines which message and response elements are supported by the NCIP server, and in what format. The DTD that the server uses is defined in tab_ncip.conf, as described above.

The Lookupverison message uses a separate DTD. The file is named ncip_version.xml

7.2 Alephe Tables

The tables described in this chapter are common to all the NCIP servers that are running on a specific installation, and are therefore defined in the alephe_tab directory.

7.2.1 tab_ncip_interface

This table defines the elements and element contents of the NCIP server responses. The structure of all NCIP responses that are generated by the server is set in this table.

Each configuration in the table is actually made up of two rows:

- A row with the value “1” in column 1. Column 2 of this row will be populated with the name of the element.
- A row with the value “2” in column 1. Column 2 of this row will be populated with the value of the element.

When column 2 is used to set the value of an element, the value can be either a quoted string, or a program name and parameters.

Below is an example in which quoted strings are used:

```
1 /UniqueUserId/UniqueAgencyId/Scheme
2 "NCIP Unique Agency Id"
1 /UniqueUserId/UniqueAgencyId/Value
2 "Exlibris Users Agency"
```

Below is an example in which a program name is used:

```
1 /UserOptionalFields/UserPrivilege/ValidFromDate
2 ncip_validfromdate:Z305
```

Below is a list of the available programs and their parameters:

ncip_addressdata:Unstructured

ncip_agencyuserprivilegetype:Value

ncip_author

ncip_datedue

ncip_electronicaddressdata

ncip_electronicaddressvalue

ncip_itemidentifiervalue

ncip_mediumtype:Scheme,"*Scheme Name*"

:Value

ncip_personalusername:Unstructured

ncip_publicationdate

ncip_publisher

ncip_title

ncip_useridentifier

ncip_userprivilegedescription

ncip_validfromdate:Z304

:Z305

ncip_validtodate:Z304

:Z305

ncip_bibitemidentifiercode: Scheme,"NCIP Bibliographic Item Identifier Code Scheme" Value

ncip_locationnameinstance:Level [1|2|3]

Value [1|2|3]

ncip_locationtype:Scheme,*"NCIP Location Type Scheme"*

Value

ncip_versionsupported

ncip_bibitemidentifier

ncip_circulationstatusvalue

ncip_bibrecordidentifiercode: Scheme,"NCIP Bibliographic Record Identifier Code Scheme" Value

ncip_bibrecordidentifier

ncip_borrower_info: Cat1 – reports the Z305-TYPE
LIBRARY – reports the patron home library, or the ADM library if no home library is defined
PROFILE – reports the patron Z305-BOR-STATUS in the home library, or in the ADM library if no home library is defined

STATUS – reports whether the patron is blocked (by PAT-DELINQ section of tab_check_circ) in the home library, or in the ADM library if no home library is defined

7.2.2 Tab_ncip_scheme

This table maps between scheme names and agency addresses. The scheme names can be used as element values in tab_ncip_interface where a scheme agency address is required.

7.3 www_server.conf

A www_server.conf variable is used to set which ADM library will be accessed by each NCIP initiator, according to the initiator's IP. The variable has the following format:

setenv www_ncip_library_[IP] [ADM library]

For example, the definition

setenv www_ncip_library_001.010.235.011 USM50

means that an initiator that is accessing the NCIP server over HTTPS from a machine with IP of 001.010.235.011 will have access only to USM50.

Note: In Aleph 21 and later this variable can have an IP with an underscore (_) instead of dots, as in the following format:

```
setenv www_ncip_library_001_010_235_011          USM50
```

This is to accommodate operating systems that do not allow environment variables to contain dots in the variable name.

7.4 Setting up Agencies

The agencies from which the ALEPH NCIP server accepts messages are used for two purposes:

- As Hold Request Pickup Locations – The 'FromAgencyId' will always be set as the pickup location of an item that has been requested via NCIP. This means that the values that are expected to arrive as FromAgencyId values must be configured in tab37 as valid pickup locations for items that are available via NCIP. For example, if an expected FromAgencyId is 'URSA' the lines in tab37 must include URSA as a valid pickup location.

  As a consequence, the values that are expected as FromAgencyId values in incoming NCIP messages should also be defined as patron IDs for pseudo-patrons. This way items may be transferred to the pickup location using ALEPH's standard mechanism for handling transfer. In the previous example, if patron with the ID URSA is defined, when an item with pickup location URSA is found it will be placed in transit to URSA by creating a loan to the patron URSA.

- When a check out, request, or accept item message is received and the patron is not an ALEPH patron, the check-out or request will be charged to a patron with the ID as in the FromAgencyId value. For example, if the FromAgencyId URSA sent a CheckoutItem message with the ID set to an ID that ALEPH does not identify, the item will be loaned to the patron URSA.

  Identifying whether or not the patron is an ALEPH patron is done in the following way:

  If the patron is identified in the message by using a UniqueUserId element, then the UniqueAgencyId will be compared with the tab_ncip.conf variable user_agency. If the patron's agency is not the agency that is identified in the tab_ncip.conf table then the patron will be considered a non ALEPH patron.

  If no UniqueUserId is present in the message then the received ID (for example, as present in the AuthenticationInput element) will be checked. If it is not authenticated with ALEPH it will be considered a non-ALEPH patron.

7.5 Home Library

ALEPH patrons that are active using NCIP must have a home library defined. The home library is where the patron will receive items that have been requested from other Item Agencies.

7.6 Creation and Retrieval of a Temporary Item

When a temporary item is created, Aleph creates a unique barcode. The barcode should be attached to the item, with which circulation activities can be performed. If, for some reason, the temporary item barcode is not attached to the item, the library can still perform all circulation activities, such as retrieval, by using the item's call number.

In order to carry out circulation activities by call number, implement the following settings in tab100 of the ADM library:

```
ITEM-BARCODE-OR-CALL-NO=Y
```

This setup enables item retrieval by call number. Aleph first attempts a retrieval by using the barcode. If that fails, Aleph makes another attempt by using the call number.

The following sections describe two possible workflows for retrieving an item by barcode. Each workflow has different settings to be implemented.

7.6.1 Temporary Item – Unique Barcode

This workflow is based on the temporary item's call number which contains an external barcode and a prefix (agency code). The external barcode together with the prefix comprises a unique identifier that can be retrieved later by staff users.

Set the tab100 parameter ITEM-BARCODE-OR-CALL-NO to `Y` to retrieve by call number. To retrieve the item, the staff types the agency code with the external barcode. Because the number is unique, only one item is retrieved.

Staff may also retrieve the item by indexing the call number (agency code + external barcode) by expanding and indexing the item's call number into the BIB record:

To expand and index the item's call number into the BIB record:

1. Set `tab_filing` in `$data_tab` of the current ILL library the following section:

```
!* NCIP Call Numbers
24   del_subfield
24   to_lower
```

2. Set `tab00.eng $data_tab` of the ILL library.

```
H NCP   IND     24 00       00       852 Call no.
...
```

3. Set `tab11_ind $data_tab` of the ILL library with the following line:

```
852##                    NCP   h
```

   This means that the BIB record field 852 is indexed as a direct index called `NCP` subfield `$$h`.

4. Set `tab_expand` in `$data_tab` of the ILL library with the following line :

```
INDEX      expand_doc_bib_852_1
```

5. Re-index the direct index by running `p-manage-05` on the ILL library.

6. Set `pc_tab_sear.lng` in `$data_tab` of the ILL library with the following lines:

```
SC  USM40               L 852 Call no. from NCIP 40      NCP
FI  USM40               L 852 Call no. from NCIP 40      NCP
```

Now the NCIP BIB record is indexed in XXX40. The BIB record contains the agency code as part of the item call number.

The BIB record is searchable and can be browsed by the item call number, which consists of the agency code in lower case as a prefix and the external barcode.

7.6.2 Temporary Item – External Barcode (Retrieval by Non Unique call-Number

This workflow is based on the temporary item's call number that only consists of the external barcode. In this case, the prefix is not added; therefore, the item call number may not be unique (The same barcode can exist from two different agencies).

By setting the tab100 parameter ITEM-BARCODE-OR-CALL-NO to `Y` the staff can scan the external barcode on the actual item and retrieve it. Note that if there is more than one matching result, the system only displays one of them. If the first retrieved barcode does not match the actual item (multiple match) it is recommended to expand and index the item's call number (agency's code) into the BIB record as described in Temporary Item – Unique Barcode on page 28.

To enable the storage of the external barcode without a prefix, set `[ill_lib (XXX40)]/data_tab/tab100` with the following parameter:

```
NCIP-CREATE-EXT-BARCODE=Y
```

In this case, the call number is created without the prefix. The default is `N`, which means that the call number is created with the prefix.

8 Running and Testing the NCIP Servers

Running the NCIP server is done by using the UTIL W\3\7\2 utility. The port number that is suggested by default can be overridden so that the server will run on a different port. The ADM library that is active when the server is launched is the library that will be used by the server. The BIB and HOL libraries will be determined according to the ADM library's relations. The NCIP server's log can be viewed in the $LOGDIR directory.

Testing the NCIP server can be done by using the UTIL/F/2/22 utility. This utility enables to edit XML files that are found in the ./alephm/tests/ncip_messages directory and to send them to a specific server, or to the default server. The XML response is received by the utility and is displayed on the screen.

Appendix A. Sample Messages and Responses

RequestItem Sample

Below is an example of a valid RequestItem service XML file:

```
<?xml version ="1.0"?>
<!DOCTYPE NCIPMessage SYSTEM "NCIP_v0_1d.dtd">
<NCIPMessage version = "1.0">
<RequestItem>

    <InitiationHeader>
        <FromSystemId>
            <Scheme datatype="string">NCIP System Ids</Scheme>
            <Value datatype="string">Exlibris From System</Value>
        </FromSystemId>

        <FromAgencyId>
            <UniqueAgencyId>
                <Scheme datatype="string">NCIP Agency Ids</Scheme>
                <Value datatype="string">Exlibris From Agency</Value>
            </UniqueAgencyId>
        </FromAgencyId>

        <ToSystemId>
            <Scheme datatype="string">NCIP System Ids</Scheme>
            <Value datatype="string">Exlibris To System</Value>
        </ToSystemId>

        <ToAgencyId>
            <UniqueAgencyId>
                <Scheme datatype="string">NCIP Agency Ids</Scheme>
                <Value datatype="string">Exlibris To Agency</Value>
            </UniqueAgencyId>
        </ToAgencyId>
    </InitiationHeader>

    <UniqueUserId>
        <UniqueAgencyId>
            <Scheme datatype="string">NCIP Agency Ids</Scheme>
            <Value datatype="string">Exlibris Item Agency</Value>
        </UniqueAgencyId>
        <UserIdentifierValue
datatype="string">00000012</UserIdentifierValue>
    </UniqueUserId>

    <UniqueItemId>
        <UniqueAgencyId>
            <Scheme datatype="string">NCIP Agency Ids</Scheme>
            <Value datatype="string">Exlibris Item Agency</Value>
        </UniqueAgencyId>
        <ItemIdentifierValue>HNX4MC</ItemIdentifierValue>
    </UniqueItemId>
```

```
    <RequestType>
        <Scheme datatype="string">NCIP Request Type Scheme</Scheme>
        <Value datatype="string">Hold</Value>
    </RequestType>

    <RequestScopeType>
        <Scheme datatype="string">NCIP Request Scope Type
Scheme</Scheme>
        <Value datatype="string">Item</Value>
    </RequestScopeType>

    <ItemElementType>
            <Scheme>NCIP User Element Type</Scheme>
            <Value>Bibliographic Item Id</Value>
    </ItemElementType>

    <UserElementType>
            <Scheme datatype="string">NCIP User Element Type</Scheme>
            <Value datatype="string">User Address Information</Value>
    </UserElementType>

    <UserElementType>
            <Scheme datatype="string">NCIP User Element Type</Scheme>
            <Value datatype="string">User Privilege</Value>
    </UserElementType>

    <UserElementType>
            <Scheme datatype="string">NCIP User Element Type</Scheme>
            <Value datatype="string">Name Information</Value>
    </UserElementType>

<ShippingInformation>
        <ShippingNote>
            <Scheme datatype="string">Shipping Note</Scheme>
            <Value datatype="string">Note for hold Request</Value>
    </ShippingNote>

</RequestItem>
</NCIPMessage>
```

RequestItemReponse Sample

Below is an example of a valid RequestItemResponse XML file:

```
<?xml version = "1.0" encoding = "UTF-8"?>
  <!DOCTYPE NCIPMessage PUBLIC "-//NISO//NCIP DTD Version 1//EN"
"http://www.niso.org/ncip/v1_0/imp1/dtd/ncip_v1_0.dtd">
    <NCIPMessage version="1.0">
      <RequestItemResponse>

        <ResponseHeader>

          <FromAgencyId>
            <UniqueAgencyId>
```

```
            <Scheme>NCIP Agency Ids</Scheme>
            <Value>Exlibris User Agency</Value>
          </UniqueAgencyId>
        </FromAgencyId>

        <ToAgencyId>
          <UniqueAgencyId>
            <Scheme>NCIP Agency Ids</Scheme>
            <Value>Exlibris Item Agency</Value>
          </UniqueAgencyId>
        </ToAgencyId>

      </ResponseHeader>

      <UniqueItemId>
        <UniqueAgencyId>
          <Scheme>NCIP Unique Agency Id</Scheme>
          <Value>Exlibris Items Agency</Value>
        </UniqueAgencyId>
        <ItemIdentifierValue>HNX4MC</ItemIdentifierValue>
      </UniqueItemId>

      <UniqueUserId>
        <UniqueAgencyId>
          <Scheme>NCIP Unique Agency Id</Scheme>
          <Value>Exlibris Users Agency</Value>
        </UniqueAgencyId>
        <UserIdentifierValue>00000012</UserIdentifierValue>
      </UniqueUserId>

     <RequestType>
        <Scheme>NCIP Request Type Scheme</Scheme>
        <Value>Hold</Value>
      </RequestType>

      <RequestScopeType>
        <Scheme>NCIP Request Scope Type Scheme</Scheme>
        <Value>Item</Value>
      </RequestScopeType>

      <ItemOptionalFields>
        <BibliographicDescription>
          <Author>Viviani della Robbia, Enrica.</Author>

          <BibliographicItemId>

<BibliographicItemIdentifier>HD999KK</BibliographicItemIdentifier>
          </BibliographicItemId>

          <BibliographicRecordId>

<BibliographicRecordIdentifier>HD999KK</BibliographicRecordIdentifier>
            <UniqueAgencyId>
              <Scheme>NCIP Unique Agency Id</Scheme>
              <Value>Exlibris Items Agency</Value>
            </UniqueAgencyId>
```

```
            </BibliographicRecordId>

            <PublicationDate>1936</PublicationDate>
            <Publisher>Firenze : G.C. Sansoni, 1936.</Publisher>
            <Title>Vita di una donna :</Title>
            <MediumType>
              <Scheme>NCIP Medium Type Scheme</Scheme>
              <Value>GG</Value>
            </MediumType>
          </BibliographicDescription>

        </ItemOptionalFields>

        <UserOptionalFields>

<NameInformation>
            <PersonalNameInformation>
              <UnstructuredPersonalUserName>oo
pp</UnstructuredPersonalUserName>
            </PersonalNameInformation>
          </NameInformation>

<UserAddressInformation>
            <UserAddressRoleType>
              <Scheme>NCIP User Address Role Type Scheme</Scheme>
              <Value>Home</Value>
            </UserAddressRoleType>

            <ValidFromDate>2005-01-01T00:00:00Z</ValidFromDate>
            <ValidToDate>2006-01-16T00:00:00Z</ValidToDate>

            <PhysicalAddress>
              <UnstructuredAddress>

                <UnstructuredAddressType>
                  <Scheme>NCIP Unstructured Address Type
Scheme</Scheme>
                  <Value>Newline-Delimited Text</Value>
                </UnstructuredAddressType>

                <UnstructuredAddressData>oo pp line 1 address line 2
address line 3 address line 4 address</UnstructuredAddressData>
              </UnstructuredAddress>

              <PhysicalAddressType>
                <Scheme>NCIP Physical Address Type Scheme</Scheme>
                <Value>Postal Address</Value>
              </PhysicalAddressType>

            </PhysicalAddress>

            <ElectronicAddress>
              <ElectronicAddressType>
                <Scheme>IANA URI Scheme</Scheme>
                <Value>Telephone</Value>
              </ElectronicAddressType>
```

```
              <ElectronicAddressData>02-6520026</ElectronicAddressData>
            </ElectronicAddress>
          </UserAddressInformation>

          <UserPrivilege>
            <UniqueAgencyId>
              <Scheme>NCIP Unique Agency Id</Scheme>
              <Value>Exlibris Users Agency</Value>
            </UniqueAgencyId>

            <AgencyUserPrivilegeType>
              <Scheme>NCIP Agency User Privilege Type Academic
Scheme</Scheme>
              <Value>Central Administration (student or
employee)</Value>
            </AgencyUserPrivilegeType>

            <ValidFromDate>2005-02-16T00:00:00Z</ValidFromDate>
            <ValidToDate>2006-05-02T00:00:00Z</ValidToDate>

            <UserPrivilegeDescription>Central Administration (student
or employee)</UserPrivilegeDescription>
          </UserPrivilege>
        </UserOptionalFields>
      </RequestItemResponse>
    </NCIPMessage>
```

AcceptItem Sample

The following is an example of a valid AcceptItem service XML file:

```
<?xml version="1.0"?>
<!DOCTYPE NCIPMessage SYSTEM "ncip_v1_0.dtd"> <NCIPMessage
version="http://www.niso.org/ncip/v1_0/imp1/dtd/ncip_v1_0.dtd">
   <AcceptItem>
      <InitiationHeader>
         <FromAgencyId>
            <UniqueAgencyId>
               <Scheme>URSA Agency IDs</Scheme>
               <Value>1930</Value>
            </UniqueAgencyId>
         </FromAgencyId>
         <ToAgencyId>
            <UniqueAgencyId>
               <Scheme>URSA Agency IDs</Scheme>
               <Value>VOYAGER</Value>
            </UniqueAgencyId>
         </ToAgencyId>
      </InitiationHeader>
      <UniqueRequestId>
         <UniqueAgencyId>
```

```
            <Scheme>URSA Agency IDs</Scheme>
            <Value>1908</Value>
         </UniqueAgencyId>
         <RequestIdentifierValue>352239</RequestIdentifierValue>
      </UniqueRequestId>
      <RequestedActionType>
         <Scheme>Action types</Scheme>
         <Value>Circulate</Value>
      </RequestedActionType>
      <UniqueUserId>
         <UniqueAgencyId>
            <Scheme>URSA Agency IDs</Scheme>
            <Value>980</Value>
         </UniqueAgencyId>
         <UserIdentifierValue>PDQ</UserIdentifierValue>
      </UniqueUserId>
      <UniqueItemId>
         <UniqueAgencyId>
            <Scheme>URSA Agency IDs</Scheme>
            <Value>Foreign Item Agency</Value>
         </UniqueAgencyId>
         <ItemIdentifierValue>bc0001101803</ItemIdentifierValue>
      </UniqueItemId>
      <ItemOptionalFields>
         <BibliographicDescription>
            <Title>Pro OpenSolaris</Title>
            <Author>Foxwell, Harry J.</Author>
            <AuthorOfComponent>Harry J. Foxwell and Christine
Tran</AuthorOfComponent>
            <TitleOfComponent>a new open source OS for Linux developers
and administrators</TitleOfComponent>
            <Edition>1</Edition>
            <PublicationDate>2009</PublicationDate>
            <Publisher>Apress</Publisher>
            <PlaceOfPublication>Berkeley, Calif.</PlaceOfPublication>
            <BibliographicItemId>
              <BibliographicItemIdentifierCode>
                <Scheme>NCIP Bibliographic Item Identifier Code
Scheme</Scheme>
                <Value>ISBN</Value>
              </BibliographicItemIdentifierCode>

<BibliographicItemIdentifier>003003003</BibliographicItemIdentifier>
            </BibliographicItemId>
            <MediumType>
              <Scheme>NCIP Medium Type Scheme</Scheme>
              <Value>Book With Audio Tape</Value>
            </MediumType>
         </BibliographicDescription>
      </ItemOptionalFields>
   </AcceptItem>
</NCIPMessage>
```

AcceptItemReponse Sample

The following is an example of a valid AcceptItemResponse XML file:

```
<?xml version = "1.0" encoding = "UTF-8"?>
  <!DOCTYPE NCIPMessage PUBLIC "-//NISO//NCIP DTD Version 1//EN"
"http://www.niso.org/ncip/v1_0/imp1/dtd/ncip_v1_0.dtd">
    <NCIPMessage
version="http://www.niso.org/ncip/v1_0/imp1/dtd/ncip_v1_0.dtd">
      <AcceptItemResponse>
        <ResponseHeader>
          <FromAgencyId>
            <UniqueAgencyId>
              <Scheme>URSA Agency IDs</Scheme>
              <Value>VOYAGER</Value>
            </UniqueAgencyId>
          </FromAgencyId>
          <ToAgencyId>
            <UniqueAgencyId>
              <Scheme>URSA Agency IDs</Scheme>
              <Value>1930</Value>
            </UniqueAgencyId>
          </ToAgencyId>
        </ResponseHeader>
        <UniqueRequestId>
          <UniqueAgencyId>
            <Scheme>NCIP Unique Agency Id</Scheme>
            <Value>Exlibris Items Agency</Value>
          </UniqueAgencyId>
          <RequestIdentifierValue>352239</RequestIdentifierValue>
        </UniqueRequestId>
        <UniqueItemId>
          <UniqueAgencyId>
            <Scheme>NCIP Unique Agency Id</Scheme>
            <Value>Exlibris Items Agency</Value>
          </UniqueAgencyId>
          <ItemIdentifierValue>1871155-10</ItemIdentifierValue>
        </UniqueItemId>
      </AcceptItemResponse>
    </NCIPMessage>
```