xConnect IDMS Architecture

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Author** | **Description** |
| 1.3 | 8/28/2012 | Ted Crane | Version 1.3 Updates |
|  | 10/11/2012 | Ted Crane |  |
| 1.5 | 10/25/2012 | Ted Crane | Version 1.5 Updates |
|  | 02/01/2013 | Ted Crane | Add High Availability content. |
| 1.6 | 03/13/2013 | Ted Crane | Version 1.6 Updates |

**Document Approvers & Sign-Off**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Approver** | **Role** | **Document Accept/Reject** |
| 09/05/2012 | Ric Merrifield | Release Manager | Accept |
| 09/05/2012 | John Stiehl | Project Manager | Accept |
| 09/05/2012 | Adam Parrish | Product Tech Manager | Accept |
| 4/11/2013 | Mark Mecham | Test Manager | Accept |

**Table of Contents**

[1 Introduction 5](#_Toc353724965)

[1.1 Description 5](#_Toc353724966)

[1.2 Purpose 5](#_Toc353724967)

[1.3 Scope 5](#_Toc353724968)

[1.4 Assumptions and Constraints 5](#_Toc353724969)

[1.5 Definitions 6](#_Toc353724970)

[2 The xConnect IDMS Environment 7](#_Toc353724971)

[3 Prerequisites 7](#_Toc353724972)

[4 Processing Sequences 8](#_Toc353724973)

[4.1 Book xBand Request Sequence 8](#_Toc353724974)

[4.2 Book xBand Sequence 9](#_Toc353724975)

[4.3 Tap To Assign Sequence 10](#_Toc353724976)

[4.4 Reassign xBand Sequence 11](#_Toc353724977)

[4.5 Public Experience Change (PxC) Sequence 12](#_Toc353724978)

[5 Messages 13](#_Toc353724979)

[5.1 Book xBand Request Message 13](#_Toc353724980)

[5.2 Book xBand Message 16](#_Toc353724981)

[5.2.1 ShortRangeTag Conversion 18](#_Toc353724982)

[5.2.2 LongRangeTag Conversion 18](#_Toc353724983)

[5.3 Change Event Message 19](#_Toc353724984)

[5.4 Callback Message 20](#_Toc353724985)

[6 IDMS Database Schema 22](#_Toc353724986)

[7 IDMS Listener Configuration 23](#_Toc353724987)

[8 High Availability 27](#_Toc353724988)

[8.1 Data Replication 27](#_Toc353724989)

[8.1.1 Publisher 27](#_Toc353724990)

[8.1.2 Subscriber 27](#_Toc353724991)

[8.2 Web application 28](#_Toc353724992)

[8.3 IDMS Listener 28](#_Toc353724993)

[9 Cache 28](#_Toc353724994)

[10 Troubleshooting 29](#_Toc353724995)

# Introduction

## Description

IDMS is a cache of guest and xBand data to support high performance lookups by other systems. In addition IDMS has been used as temporary stand in for SF/OneView during demos and tests.

IDMS is comprised of two components.

* IDMS RESTFul web services, which allow other application to retrieve guest, xBand, and celebration data. The available web services are documented in the IDMS Interface Control document.
* IDMS Listener, which consumes messages from several JMS topics and writes data to the IDMS data store using the IDMS RESTFul web services. The IDMS Listener is one service provided by the xConnect JMS Listener. Services that listen to other JMS messages are documented in the 900-0077 Rev 1.5 JMS Listener Architecture document.

## Purpose

Description of the system architecture for the xConnect IDMS Listener software that reads and messages transmitted from xBMS and caches information about the relationship between guest and the media used by the guest to interact with the xConnect system.

IDMS will be the source for other xConnect systems in the LDU to retrieve xBMS Link ID. This ID will be used by xConnect in messaging guest information to systems outside xConnect.

## Scope

This is an early draft of the xConnect IDMS Listener software, and does not cover all functionality to be provided in the production system.

## Assumptions and Constraints

A Progress Software Sonic MQ Enterprise Service Bus has been installed and configured in a location that can be connected to by the IDMS Listener software.

A SQL Server 2008 R2 Server has been installed and configured with the appropriate xBRMS database schema in a location that can be connected to by the IDMS Listener software.

xBMS is operating and publishing the expected messages on the SonicMQ Enterprise Service Bus.

The IDMS web services are available to retrieve guest information.

## Definitions

For this draft, the following acronyms are used:

|  |  |
| --- | --- |
| **Term** | **Definition** |
| xConnect | Code, scripts, APIs, and database schemas which comprise the unifying messaging, management, and reporting software which ties the hardware together into a coherent solution |
| xBand | RF-enabled wristband |
| xTP | Experience TouchPoint, a Disney-themed short range HF reader or "touch" device |
| DAP | Disney Access Portal, a free-standing xTP configured with a biometric reader at park entry or as a sole touch point stanchion for access to attractions |
| xBR | Long range RFID reader with uni- or omni-directional antennae |
| xBRC | xBand Reader Controller which manages xBRs, xTPs, and DAP devices |
| xBMS | xBand Management System code and database, which is the system of record for media, i.e. cards and bands, and the association between media and Guests via link ID architecture. |
| Media | xBands, cards or other devices that store a public or secure ID that are read by the xConnect system. |
| BOG | Be Our Guest restaurant in Fantasyland. |
| DME | Disney’s Magical Express. |
| SF-OV | Strategic Functionality and One View. |

# The xConnect IDMS Environment



Figure 1 xConnect IDMS Environment

# Prerequisites

This section addresses network, hardware, and software prerequisites to support the xConnect IDMS Listener software. The xConnect system has been installed following the instructions outlined in the *NGE Alpha Lab Installation Guide*.

# Processing Sequences

## Book xBand Request Sequence



Figure 2 Book xBand Request Sequence

The Book xBand Request sequence is initiated by a travel plan event, typically a file import of reservation data. The reservation data is provided and imported in batches.

From this data xBMS generates a Book xBand Request message on the Enterprise Service Bus.

The IDMS Listener uses the reference ID in this message to call to the xBMS Provide xBand Request Details web service to retrieve additional data about the Book xBand Request.

For additional information on both the Book xBand Request message and the response to the Provide xBand Request Details, see the [Book xBand Request Message](#_Book_xBand_Request_1) section in this document.

In production the IDMS Listener will listen to both the XBMS.XBANDREQUEST and XBMS.IDMS.XBANDREQUEST topics (shown as JMS Topic in the sequence diagram). The same message structure will be transmitted on both topics. The XBMS.XBANDREQUEST topic will contain messages about guests imported into xBMS from DME and Dreams reservations systems. The XBMS.IDMS.XBANDREQUEST topic will be used for guests specifically imported for the BOG restaurant.

The Provide xBand Request Details response can contain multiple guests, so the Create Guest message between the IDMS Listener and IDMS may occur multiple times for each message received on the JMS Topic.

If processing of the message fails, the message will be stored in memory and then processing retried. The number of retries and the interval between retries are configurable settings. After the number of retries has been exceeded, an error message will be written to the log file.

## Book xBand Sequence



Figure 3 Book xBand Sequence

The Book xBand sequence is initiated by a Fulfilled Band event, typically a file import of vendor fulfillment data.

From the data received from the fulfillment vendor, xBMS then generates a Book xBand event on the Enterprise Service Bus. This message indicates that an xBand needs to be associated to a guest.

The IDMS Listener will first check if the guest exists in IDMS by the provided guest type and guest id. If the guest doesn’t exist, the message is ignored, since the guest should already be in xBMS, and have been created in IDMS through the Book xBand Request Sequence. For situation where the guest doesn’t exist, see the [Tap To Assign Sequence](#_Tap_To_Assign).

If the guest exists, then the xBand information included in the message is used to create the xBand data in IDMS. Then the association between the xBand and the guest is created in IDMS.

If the guest has a gxp-link-id in IDMS, then the IDMS Listener will notify the FastPass+ about the new band associated to the guest. The IDMS Listener will send a notification that the band is being associated.

## Tap To Assign Sequence



Figure 4 Tap To Assign Sequence

The Tap To Assign Sequence represents the provisioning of xBands for use in the BOG restaurant to assume the role of a Puck or Cast Member. The Puck represents a guest that doesn’t have a personal band, and the xBand is enclosed in a container given to the guest when they check in. xBands used for the Cast Member role attached to a device used by a cast member such as busser or server.

The sequence begins with an application calling the xBMS Tap To Assign web service endpoint. This call will generate a Book xBand Message. The processing of this message is similar to the Book xBand Sequence, but in the case of the Tap To Assign Sequence, the guest does not already exist in IDMS. An anonymous guest is created in IDMS, with the specified guest type and guest id. This sequence also assumes the xBand needs to be created. xBands that have been previously associated with a guest will be handled in the [Reassign xBand Sequence](#_Reassign_xBand_Sequence). Once the xBand has been created in IDMS, an association to the anonymous guest will be created. Unlike the Book xBand Sequence, FastPass+ will not be notified of this association, as the guest will not have a gxp-link-id.

## Reassign xBand Sequence



Figure 5 Reassign xBand Sequence

Prior to xBands being shipped to guests, guests that pre order a meal in the BOG restaurant will need to be associated to one of the existing xBands being used in the Puck role. To accomplish this, an application will call the xBMS Modify xBand Details web service to transfer the xBand to the guest. xBMS will generate a Book xBand Message on the Enterprise Service Bus. Unlike the Book xBand Sequence or Tap To Assign Sequence, both the guest and the xBand will already exist in IDMS. So data for both the guest and the xBand will be retrieved from IDMS. This data will be used to remove the existing association between the anonymous guest and xBand in IDMS and create a new association between the actual guest and xBand in IDMS. Then the FastPass+ system is informed of the ownership change on the IDMS.NOTIFIER.GXP queue.

Once the meal has been served to the guest, the Puck will be removed from their table, and reassigned back to the anonymous guest. Then the Puck can be reused in the restaurant for guests that haven’t pre ordered.

## Public Experience Change (PxC) Sequence



Figure 6 Public Experience Change (PxC) Sequence

The Publish Experience Change Sequence allows the IDMS cache of identifiers to stay in sync with the SF/OneView, which is the system of record for the guest identifiers IDMS is caching. The IDMS Listener will listen for a Change Event message on the SF.OV.EXPERIENCEASSOCIATION topic and call the Provide Guest Identifiers IDMS web service to retrieve the existing guest identifiers stored in IDMS. These identifiers will be compared to the identifiers received in the Change Event message, and any guest identifiers not found in IDMS will be added. Only identifiers associated with a guest, and not those associated with a guest’s xBand, will be added to IDMS. IDMS will store all xBand data from xBMS instead.

# Messages

This section documents the messages the read and processed by the xConnect IDMS Listener system processes. Sample events provided from xBMS are included to help document the flow of data through the xConnect IDMS Listener system.

If processing of either the Book xBand Request Message or Book xBand Message fails, the message will be stored in memory and then processing retried. The number of retries and the interval between retries are configurable settings. After the number of retries has been exceeded, an error message will be written to the log file.

## Book xBand Request Message

The Book xBand Request message is documented on the xBMS wiki page:

<https://wiki.nge.wdig.com/display/NGE/xBMS+Event+-+Book+xband+Request>

A sample Book xBand Request Message:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<businessEvent>

<guestIdentifier></guestIdentifier>

<eventType>BOOK</eventType>

<subType></subType>

<location>XBMS.XBANDREQUEST</location>

<timestamp>March 28, 2011 9:39:01 PM UTC</timestamp>

<referenceId>EE692F11-8982-47EB-9717-24EFB9C15D35</referenceId>

<payload><![CDATA[]]></payload>

<correlationId>21ffefd0-c279-46d8-b978-fe268868b509</correlationId>

</businessEvent>

From this message, the referenceId value is used to retrieve the details of the request, as documented on the xBMS wiki.

<https://wiki.nge.wdig.com/display/NGE/xBMS+Provide+xBand+Request>

This endpoint provides a response in JSON, as shown in the following example:

{

"customizationSelections" : [

{

"xbandOwnerId" : "0E164AC1-6230-4DC5-A5D5-C5F30F60C43F",

"self" : "/customization-selections/0E164AC1-6230-4DC5-A5D5-C5F30F60C43F",

"xband" : "/xband/0E164AC1-6230-4DC5-A5D5-C5F30F60C43F",

"birthDate" : "1975-02-01T05:00:00Z",

"firstName" : "Jane",

"guestId" : 10000,

"guestIdType" : "transactional-guest-id",

"primaryGuest" : true,

"lastName" : "Doe",

"bandProductCode" : "B11111",

"bandAccessories" : [

{

"bandAccessoryCode": "100000",

"bandAccessoryType": "100000"

}

],

"printedName" : "Jane's Band",

"customizationSelectionId" : "0E164AC1-6230-4DC5-A5D5-C5F30F60C43F",

"createDate" : "2010-07-26T05:00:00Z",

"updateDate" : "2010-07-26T05:00:00Z",

"entitlements" : [ "STANDARD", "LEVEL\_2\_5" ],

"qualifyingIds" : [

{

"qualifyingId": 1234579,

"qualifyingIdType": "travel-component-id"

},

{

"qualifyingId": 1234578,

"qualifyingIdType": "travel-component-id"

},

{

"qualifyingId": 4387621045,

"qualifyingIdType": "bundle-id"

},

{

"qualifyingId": 4387621048,

"qualifyingIdType": "bundle-id"

},

],

"confirmedCustomization" : false,

},

{

"xbandOwnerId" : "D7C0EF05-5078-4ECD-A3DF-D53A65E0A75C",

"self" : "/customization-selections/8D60B701-22EA-487D-8950-D185B46D4EDC",

"xband" : "/xband/8D60B701-22EA-487D-8950-D185B46D4EDC",

"birthDate" : "1980-01-01T05:00:00Z",

"firstName" : "Joey",

"guestId" : 10001,

"guestIdType" : "transactional-guest-id",

"primaryGuest" : false,

"lastName" : "Tsai",

"bandProductCode" : "B11111",

"bandAccessories" : [

{

"bandAccessoryCode": "100000",

"bandAccessoryType": "100000"

}

],

"printedName" : "Joey's band",

"customizationSelectionId" : "8D60B701-22EA-487D-8950-D185B46D4EDC",

"createDate" : "2010-07-26T05:00:00Z",

"updateDate" : "2010-07-26T05:00:00Z"

"entitlements" : [ "NONE" ],

"qualifyingIds" : [

{

"qualifyingId": 1234578,

"qualifyingIdType": "travel-component-id"

}

],

"confirmedCustomization" : false,

},

],

"requestAddress" :

{

"confirmedAddress" : false,

"phoneNumber" : "867-5309",

"address" : {

"address1" : "#4 Privet Drive",

"city" : "Atlanta",

"country" : "US",

"postalCode" : "30328",

"state" : "GA"

}

},

"resortReservations": [

{

"arrivalDate" : "2010-09-26T04:00:00Z",

"departureDate" : "2010-10-09T04:00:00Z",

"facilityId" : 280010388,

"travelSegmentId": 1234578,

"travelComponentId": 87654321

},

{

"arrivalDate" : "2010-09-26T04:00:00Z",

"departureDate" : "2010-10-09T04:00:00Z",

"facilityId" : 280010400,

"travelSegmentId": 1234579,

"travelComponentId": 87654320

}

],

"shipment" : {

"method": "PRIMARY\_GUEST\_ADDRESS\_BEST",

"carrier" : "UPS",

"carrierLink" : "http://www.ups.com/WebTracking/track",

"shippingDate" : "2010-08-27T04:00:00Z",

"trackingNumber" : 66,

"address" : {

"address1" : "#4 Privet Drive",

"city" : "Atlanta",

"country" : "US",

"postalCode" : 30328,

"state" : "GA"

}

},

"state" : "COMPLETED",

"createDate" : "2010-07-26T05:00:00Z",

"updateDate" : "2010-07-26T05:00:00Z",

"customizationEndDate" : "2010-08-26T05:00:00Z",

"xbandRequestId" : "B2512223-7314-46EE-9687-8B6159ECAD08",

"self" : "/xband-requests/B2512223-7314-46EE-9687-8B6159ECAD08",

"reorder" : "/xband-requests/B2512223-7314-46EE-9687-8B6159ECAD08/reorder",

"options": "/reorder-options/458414CE-602C-4ECC-B01D-81D4BE7EC29C",

"order": "/orders/85E5B54C-3727-4A57-8B13-5570D4B3657D",

"acquisitionId" : "100077",

"acquisitionIdType" : "travel-plan-id",

"acquisitionStartDate" : "2010-08-26T05:00:00Z",

"acquisitionUpdateDate" : "2012-01-26T05:00:00.568Z",

}

For each guest item in the customization selections list, the transactional guest ID (or what every guest id type is specified) is first used to check if the guest already exists. If the guest doesn’t exist, the highlighted fields are used to create a new guest in IDMS.

## Book xBand Message

The Book xBand message is documented on the the following xBMS wiki page:

<https://wiki.nge.wdig.com/display/NGE/xBMS+Event+-+Book+xband+Request>

A sample Book xBand Message:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<businessEvent>

<guestIdentifier></guestIdentifier>

<eventType>BOOK</eventType>

<subType></subType>

<location>XBMS.XBAND</location>

<timestamp>March 28, 2011 9:39:01 PM UTC</timestamp>

<referenceId>EE692F11-8982-47EB-9717-24EFB9C15D35</referenceId>

<payload><![CDATA[]]></payload>

<correlationId>21ffefd0-c279-46d8-b978-fe268868b509</correlationId>

</businessEvent>

From this message, the referenceId value is used to retrieve the details of the request from the Provide xBand Details endpoint, as documented on the following xBMS wiki page:

<https://wiki.nge.wdig.com/display/NGE/xBMS+Provide+xBand+Details>

JSON, like the following example, is returned:

{

"assignmentDateTime" : "2011-02-22T04:09:47Z",

"externalNumber" : 4320000000000000,

"printedName" : "Carol Xband3",

"productId" : "B11012",

"publicId" : 1234,

"secondaryState" : "ORIGINAL",

"secureId" : 66,

"self" : "/xband/0192B866-85BD-4369-9E85-84237315A110",

"shortRangeTag" : 180166,

"state" : "ACTIVE",

"guestId" : "0915EFB4-3B7D-45B4-93FD-D7145941D484",

"guestIdType" : "swid",

"xbandId" : "0192B866-85BD-4369-9E85-84237315A110",

"xbandRequest" : "/xband-requests/6315A9F3-C19C-4530-8337-A73AA21C33D3",

"xbandOwnerId" : "6422FA29-6BDB-4831-9F5F-6A2A165E3FA1",

"options" : "/xband-options/0192B866-85BD-4369-9E85-84237315A110"

"history" : "/xband-history/0192B866-85BD-4369-9E85-84237315A110"

}

The highlighted values are used to create a band in IDMS. The xbandOwnerID is used to associate the guest and the band in IDMS.

xBand data can be retrieved from IDMS using xxxxx, the following example, using publicId returns the following data (guest data excluded).

{

"state": "Active",

"publicId": "2063557",

"productId": "170136025267",

"xbandId": 36067815,

"bandType": "Guest",

"bandId": "170136025267",

"printedName": "Beth",

"xbandRequestId": "170136025267",

"shortRangeTag": "0453872A892B80",

"longRangeTag": "00001F7CC5",

"secondaryState": "Active"

}

When processing a Book xBand message, the following mapping is performed before storing the data in IDMS:

|  |  |
| --- | --- |
| **IDMS** | **xBMS** |
| state (“active” or “inactive”) | state |
| publicId | publicId |
| productId (not used, to be removed) |  |
| xbandId | IDMS key for xBand |
| bandType | bandRole or “Guest” if not specified |
| bandId | externalNumber |
| printedName | printedName or “Printed Name “ + publicId if not specified |
| xBandRequestId (not used, to be removed or updated to match xBMS) |  |
| shortRangeTag | shortRangeTag (Converted) |
| longRangeTag | publicId (converted) |
| secondaryState (always matches state) |  |

Both publicId and shortRangeTag are stored in xBMS as integer values, but the ID values provided to the xBRC by the xTPs and xBRs are hexadecimal representations of the values read from the physical media represented by these ids in xBMS.

### ShortRangeTag Conversion

The value of shortRangeTag from xBMS is first converted from a decimal integer to a hexadecimal string, i.e. 1341700931 would be converted to “0000004FF8BB43”. The hexadecimal value is always padded with zeroes to make a 14 character string. The byte order is then reversed. Each byte is represented by two characters, so the result of the example hexadecimal string would result in “43BBF84F000000” being stored as shortRangeTag in IDMS.

### LongRangeTag Conversion

Similar to shortRangeTag, longRangeTag must also be converted to hexadecimal, but the reordering of the bytes is not necessary. For example a publicId of 1341700931 would be converted to longRangeTag, in IDMS, as “004FF8BB43”, as only 10 characters are used for the longRangeTag.

## Change Event Message

The Change Event Message is received from PxC and indicates some aspect of the guest has changed. Not all guest experience changes are sent to IDMS.

For more information on the Change Event Message can be found on the NGE wiki:

<https://wiki.nge.wdig.com/display/NGE/SF+Packaging+-+Publish+Guest+Experience+Recognition+Changes>

The Change Event Message arrives in the standard Business Event XML format, with a JSON payload. The following, with some unused values removed, is an example of the JSON received in the Change Event Message:

{

"timestamp": "2002-02-13T12:00:00z",

"type": "association|combination|ownership",

"changeNotificationTrigger": "resortReservation",

"changeNotificationTriggerId": "resortReservationId123",

"resultingGuestIdentifiers": [],

"previousGuestIdentifiers": [],

"owner": {

"links": {

"self": {

"href": "<assembly-context>/guest/xid001/profile"

}

},

"guestIdentifiers": [

{

"type": "xid",

"value": "xid001"

},

{

"type": "swid",

"value": "3F68BA63-189F-4E25-AEBE-A3E92C0FA96C"

},

{

"type": "gxp-link-id",

"value": "gxp-link-id001"

}

],

"entitlements": {

"xpasses": [],

"resortReservations": [],

"admissionReservations": [],

"celebrations": [],

"dmeReservations": []

},

"xbands": {

"self": {

"href": "<assembly-context>/guest/xid001/managed-xbands"

},

"entries": []

}

}

}

The only information the IDMS Listener processes from the Change Event Message is are the guest identifiers. A lookup for the guest identifier is performed to determine if the type and value have already been stored in IDMS. If not, the guest identifier is added. Not all identifiers received from PxC are stored in IDMS. If an unknown identifier is received, no attempt is made to retrieve or add the guest identifier to IDMS. The supported identifiers types can be found at the guest/locators endpoint in IDMS.

In addition the Change Event Message has three properties in the header:

* CORRELATION\_ID
* SOR\_NAME
* orchestrationId

The correlation id and orchestration id settings need to be preserved for each message so they can be include in the header of the Callback Message.

## Callback Message

The IDMS is required to send a Callback Message to PxC to acknowledge receipt of the Change Event Message. The Callback Message has a JSON payload, without the Business Event XML wrapper. The JSON is in the following form:

{ "sor" : "IDMS",

"status": "FAILURE|SUCCESS|NOT\_APPLICABLE",

"orchestrationId" : "706A4E850E70469B9A75FD81F5B0472C"

}

The value for sor will always be “IDMS”, as IDMS is the system of record for the message. The message status will be set to “SUCCESS” if the Change Event Message is successfully processed or “FAILURE” if it is not.

The orchestrationId is set to the value preserved from the Change Event Message.

The Callback Message must also have the following properties set in the message header:

* CORRELATION\_ID - same as the value received from the Change Event Message header.
* SOR\_NAME – Always “IDMS”.
* orchestrationId - same as the value received from the Change Event Message header.

# IDMS Database Schema



Figure 7 IDMS Database Schema

The IDMS database is the persistence mechanism for the caching guest and xBand data retrieved from xBMS messages and web service endpoints.

# IDMS Listener Configuration

The IDMS Listener depends on both the environment.properties file located in /etc/nge/config, as well as the config table in the xBRMS database.

The environment.properties file contains the information the IDMS Listener needs to connect to the database to read additional configuration information. All passwords are also stored in the environment.properties file, so they can be encrypted and stored in the file. The standard NGE decryption process is used to decrypt the passwords so they can be used by the IDMS Listener. The configuration of common JMS Listener settings and the settings specific to the IDMS Listener are covered in this document. Other JMS Listener settings are covered in the 900-0077 Rev 1.5 JMS Listener Architecture document.

Common JMS Listener configuration properties:

|  |  |
| --- | --- |
| **Property Name** | **Description** |
| nge.xconnect.jmslistener.cacheExpriationMinutes | Optional. Number of minutes the data retrieved from IDMS is cached (not used by the IDMS listener). Set to 60, if not specified. |
| nge.xconnect.jmslistener.cacheSize | Optional. Maximum number of entries in the cache (not used by the IDMS listener). Set to 10,000, if not specified. |
| nge.xconnect.jmslistener.usessl | Optional. Indicates whether or not the JMS Listener web server uses SSL or not. Set to false, if not specified. |
| nge.xconnect.jmslistener.httpPort | Optional. The port used for HTTP communication with the web server. Set to 8081, if not specified. |
| nge.xconnect.jmslistener.httpsPort | Optional. The port used for HTTPS communication with the web server. Set to 8009, if not specified. |
| nge.eventserver.jmslistener.uid | Required. User ID the JMS Listener uses to connect to the broker for transmitting discovery messages. |
| nge.eventserver.jmslistener.uid | Required. The password the JMS Listener uses to connect to the broker for transmitting discovery messages. |
| nge.xconnect.jmslistener.discovery.topic | Required. The topic the JMS Listener uses to transmit discovery messages. |
| nge.xconnect.jmslistener.ssl.keystore.server | Required. |
| nge.xconnect.jmslistener.ssl.keystore.server.pwd | Required. |
| nge.xconnect.jmslistener.ssl.keystore.client | Required. |
| nge.xconnect.jmslistener.ssl.keystore.client.pwd | Required. |
| nge.xconnect.jmslistener.retry.attempts | Optional. Number of attempts to retry processing a message. Default value of 3. |
| nge.xconnect.jmslistener.retry.msec | Optional. Number of milliseconds to wait until processing of a message is retried. Default value of 10,000 (10 seconds). |
| nge.xconnect.jmslistener.processing.interval | Optional. Interval, in milliseconds, that the queue of messages is processed. Default value of 500. |
| nge.xconnect.jmslistener.ping.interval | Optional. Number of seconds between pings to the JMS broker. Helps in detecting lost connections. Default value of 5. |
| nge.xconnect.jmslistener.socket.timeout | Optional. Number of milliseconds before a socket timeout occurs when connected to the broker. Default is 5000. |

Properties specific to the IDMS Listener configuration:

|  |  |
| --- | --- |
| **Property** | **Description** |
| nge.eventserver.idmslistener.uid | Required. User ID the JMS Listener uses to connect to the broker for listening to messages from xBMS, and transmitting messages to the FastPass+ system and PxC. |
| nge.eventserver.idmslistener.pwd | Required. Password the JMS Listener uses to connect to the broker for listening to messages from xBMS, and transmitting messages to the FastPass+ system and PxC. |
| nge.xconnect.idmslistener.sharedconfigcount | Optional. Number of shared connections to create when connecting to the JMS Broker. Defaults to 10 when not specified. |
| nge.xconnect.idmslistener.jms.retryperiod | Optional. Number of milliseconds to wait to retry connecting when a connection to the JMS Broker is lost or cannot be established. Set to 10,000 if not specified. |
| nge.xconnect.idmslistener.idms.rooturl | Required. The root of the URL used to make web service calls to IDMS. |
| nge.xconnect.idmslistener.xbms.rooturl | Required. The root of the URL used to make web service calls to xBMS. |
| nge.xconnect.idmslistener.xbms.xbandrequest.topic | Required. Topic the IDMS Listener listens for general xBand Request messages. |
| nge.xconnect.idmslistener.xbms.xband.topic | Required. Topic the IDMS Listener listens for general xBand messages. |
| nge.xconnect.idmslistener.idms.xbandrequest.topic | Required. Topic the IDMS Listener listens for BOG specific xBand Request messages. |
| nge.xconnect.idmslistener.xbms.xband.topic | Required. Topic the IDMS Listener listens for BOG specific xBand messages. |
| nge.xconnect.idmslistener.pxc.receivequeue | Required. The name of the queue experience Change Event Messages are received on. |
| nge.xconnect.idmslistener.pxc.callbackqueue | Required. The name of the queue experience Change Event Messages are acknowledge on. |
| nge.xconnect.idmslistener.notification.queue | Required. The name of the queue that band ownership change notifications are made on. |

# High Availability

## Data Replication

To support a resilient environment that support high performance retrieval of information about guest, xBand, and guest identifiers, SQL Server replication has been employed to maintain data synchronization between a master IDMS database in the data center and their counterpart in the LDU. Specifically, transactional replication is being used. For more information SQL Server replication see <http://msdn.microsoft.com/en-us/library/ms151198%28v=sql.105%29.aspx>.

### Publisher

The IDMS Listener will write data from xBMS and SF-OV (via PxC) to the IDMS database in DISC. The IDMS database in DISC will be used at the SQL Server replication publisher. Once data is written to the publisher database it will be replicated to the subscribers.

### Subscriber

The IDMS subscriber databases are read only copies of in both structure and stored data of the publisher. They are designed to be a high performance frontline cache of data, primarily to support the resolution of band ids to a guest by xBRC when receiving long range reads.

Currently the publisher and subscriber databases contain the same data. Once actual data usage patterns have been established, the data replicated from publisher to subscriber can be modified to only include the data needed by the xBRC. This subset of data could be based upon date, or other factors. Limiting the data sent to the subscriber would have the benefits of reducing storage requirements for the subscribers, maintaining high performance lookups, and limiting the network traffic required to replicate the data.

## Web application

In the DISC and each LDU environment a pair of web servers that connect to the IDMS database in the appropriate environment. The web servers will be installed behind a VIP and be accessed in a round robin fashion.

## IDMS Listener

Redundant copies of the IDMS Listener will be installed on separate servers. When connecting to the broker the listener will use a shared configuration. The broker will then distribute messages to the topic or queue that was subscribed to across both servers. If a single server fails, or needs to be shut down for maintenance, all the messages will be sent by the broker to the remaining operational server.

# Cache

The cache is an optional component that may be used to reduce network load between the IDMS server and database. It stores, in memory, results from the database up to some configured maximum number of objects. Objects retrieved that would exceed the maximum number of allowed objects are not stored in the cache. The cache may configured to be cleared once a day or never at all.

The status message will present cache details when the cache is enabled:

<statusMessage>Guest objects 1023/1000000

Band objects 2098/3000000

Cache is enabled</statusMessage>

The following is an example configuration from /etc/nge/config/environment.properties that allows for caching a maximum of one million guests, three million bands, and is cleared each day at 2:00am:

nge.xconnect.idms.cache.enable=true

nge.xconnect.idms.cache.maxGuests=1000000

nge.xconnect.idms.cache.maxBands=3000000

nge.xconnect.idms.cache.resetTime=02:00

|  |  |
| --- | --- |
| **Property** | **Description** |
| nge.xconnect.idm.cache.enable | Set to ‘true’ to turn on caching. Default: false |
| nge.xconnect.idm.cache.maxGuest | The maximum number of guest objects to hold in memory. Default: 1000000 |
| nge.xconnect.idm.cache.maxBands | The maximum number of band objects to hold in memory. Default: 3000000 |
| nge.xconnect.idm.cache.resetTime | Time of day (HH:mm, 24-hr clock) to clear cache. Default: “” |

# Troubleshooting

Please refer to the *xConnect System* *Troubleshooting.docx* for troubleshooting instructions.