

API & Schema Reference Manual

BillerXpert 4.6 B2B Edition

May 2002

Copyright © 2002 Sun Microsystems, Inc. All rights reserved.

Sun, Sun Microsystems, the Sun logo, iPlanet, Java and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of the Sun Microsystems, Inc. and its licensors, if any.

THIS DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2002 Sun Microsystems, Inc. Tous droits réservés.

Sun, Sun Microsystems, et Sun logo, iPlanet, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et d'autre pays. iPlanet et the iPlanet logo sont des marques de fabrique ou des marques déposées de Sun Microsystems aux Etats-Unis et d'autre pays.

Le produit décrit dans ce document est distribué selon des conditions de licence qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ni de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit sans l'autorisation écrite préalable de Sun Microsystems, Inc. et, le cas échéant, de ses bailleurs de licence.

CETTE DOCUMENTATION EST FOURNIE “EN L'ÉTAT”, ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Contents

List of Tables	13
About This Document	19
Audience	19
Required Documents	20
What's in This Document	20
Documentation Conventions	21
The Document Online	22
Related Documentation	22
Product Support	22
Introduction to BillerXpert B2B Classes and Database Schemas	25
Component Business Object Interface Classes	25
Component Objects	25
Base and Core Classes	26
Billing Classes	28
Membership Classes	29
Event Classes	30
Database Tables	30
LDAP Schema	32
Enumerated Domains	32

Part 1 BillerXpert B2B Classes 33

Chapter 1 Base and Component Business Object Interfaces	35
Overview	35
ACGUIDException Class Reference	38
Class Definition	38
Constructor	38
ACGUIDGeneration Class Reference	38
Class Definition	39
Methods	39
ActivityLogger Class Reference	40
IActivityLogger Class Reference	41
Class Definition	41
Methods	41
ActivityLoggerAL Class Reference	41
Class Definition	42
Methods	42
ActivityLoggerMgr Class Reference	43
Class Definition	43
Methods	43
AuthToken Class Reference	43
Class Definition	44
Methods	44
Fields	51
BaseAppLogic Class Reference	51
Class Definition	52
Constructor	52
Fields	52
BXActivities Class Reference	53
Class Definition	53
Methods	53
BXActivity Class Reference	54
BXActivity Class Definition	54
BXActivity Class Methods	54
BXActivityCheck Class Reference	59
Class Definition	59
Methods	59
BXBaseDef Class Reference	61
Class Definition	61
Methods	61
BXConfig Class Reference	62
Class Definition	62
Methods	62

BXConverter Class Reference	65
Class Definition	66
Methods	66
BXDatetime Class Reference	67
Class Description	67
Methods	67
BXDBContext Class Reference	72
IBXDBContext Class Reference	72
Class Definition	72
Methods	73
BXDBDef Class Reference	75
BXDef Class Reference	75
BXEntityObject Class Reference	75
Class Definition	75
Fields	75
Methods	76
BXEnumDef Class Reference	79
Class Definition	79
Methods	79
BXError Class Reference	81
Class Definition	81
Fields	81
BXException Class Reference	82
Class Definition	82
BXFileId Class Reference	82
Class Definition	82
Methods	83
BXFilter Class Reference	85
Class Definition	85
Methods	85
BXGlobal Class Reference	92
Class Definition	92
Methods	92
Fields	94
BXJDBCHelper Class Reference	95
Class Definition	95
Methods	95
BXMessages Class Reference	104
Class Definition	105
Methods	105
BXObjectMgr Class Reference	106
Class Definition	106
Methods	106

BXObjMap Class Reference	107
Class Definition	107
Methods	107
BXOrder Class Reference	109
Class Definition	109
Order Specification Constants	109
Methods	109
BXStatus Class Reference	113
Class Definition	113
Constants	113
Constructor	115
Methods	116
BXStopwatch Class Reference	121
Class Definition	121
Methods	121
BXTableDef Class Reference	124
Class Definition	124
Methods	124
IBXEntity Class Reference	127
Class Definition	128
Methods	128
SearchResultSet Class Reference	134
Class Definition	134
Constructor	134
Fields	134
 Chapter 2 Billing Classes Component Business Object Interfaces	137
Overview	137
BXBillerCustAcct Class Reference	139
IBXBillerCustAcct Class Reference	139
Class Definition	139
Methods	139
BXBillingContext Class Reference	162
Class Definition	162
Constructor	163
BXBillingEntity Class Reference	164
IBXBillingEntity Class Reference	164
Class Definition	164
Methods	165
BXBillingMgr Class Reference	167
Class Definition	168
Methods	168
BXCardInfo Class Reference	172

IBXCardInfo Class Reference	172
Class Definition	172
Methods	172
BXContainer Class Reference	184
IBXContainer Class Reference	184
Class Definition	185
Methods	185
BXDetail Class Reference	194
IBXDetail Class Reference	194
Class Definition	194
Methods	194
BXDocument Class Reference	197
IBXDocument Class Reference	198
Class Definition	198
Methods	198
BXEventLog Class Reference	215
IBXEventLog Class Reference	215
Class Definition	216
Methods	216
BXEventLogDetail Class Reference	221
Class Description	221
Methods	221
BXExtensionDesc Class Reference	223
IBXExtensionDesc Class Reference	223
Class Description	223
Methods	224
BXLineItem Class Reference	225
IBXLineItem Class Reference	226
Class Definition	226
Methods	226
BXMessage Class Reference	239
IBXMessage Class Reference	239
Class Definition	240
Methods	240
BXOFXSignOn Class Reference	251
IBXOFXSignOn Class Reference	251
Class Description	252
Methods	252
BXPayHistory Class Reference	253
IBXPayHistory Class Reference	254
Methods	254
BXPaySchedule Class Reference	267
IBXPaySchedule Class Reference	268

Class Definition	268
Methods	268
BXSummary Class Reference	281
IBXSummary Class Reference	282
Class Definition	282
Methods	282
Chapter 3 Membership Component Business Object Interfaces	305
Overview	305
Membership Classes	306
BXAddress Class Reference	306
IBXAddress Class Reference	307
Class Definition	307
Methods	307
BXBillerProfile Class Reference	312
IBXBillerProfile Class Reference	312
Interface Definition	313
Methods	313
BXCustomerProfile Class Reference	336
IBXCustomerProfile Class Reference	337
Interface Definition	337
Methods	337
IBXFinancialAccount Class Reference	362
IBXFinancialAccount Class Reference	362
Interface Definition	363
Methods	363
BXMembershipContext Class Reference	367
Class Definition	367
Log Message Level Constants	367
Constructor	368
Methods	368
BXMembershipEntity Class Reference	371
IBXMembershipEntity Class Reference	371
Interface Definition	372
Methods	372
BXMembershipMgr Class Reference	373
Class Definition	373
Methods	373
BXPaymentType Class Reference	377
IBXPaymentType Class Reference	377
Interface Definition	377
Methods	378

Chapter 4 Event Component Business	383
Object Interfaces	383
Overview	383
BXEvent Class Reference	384
IBXEvent Class Reference	384
Interface Definition	384
Methods	385
BXEventContext Class Reference	389
Class Definition	389
Constructors	389
BXEventEntity Class Reference	390
IBXEventEntity Class Reference	391
Interface Definition	391
Methods	391
BXEventHandler Class Reference	392
IBXEventHandler Class Reference	392
Class Description	392
Methods	393
BXEventManager Class Reference	399
Class Definition	399
Methods	399
BXEventQueue Class Reference	401
IBXEventQueue Class Reference	401
Interface Definition	401
Methods	402
BXHandler Class Reference	410
IBXHandler Class Reference	410
Class Definition	410
Methods	411

Part 2 BillerXpert Database Schemas 415

Chapter 5 Billing Schema	417
Billing Schema	418
Relationship Between Tables	418
Listing of Tables	419
Event Tables	419
BXEvent Table	420
BXHandler Tables	420
BXEventHandler Table	421
BXEventQueue Table	422
Activity Tracking Tables	423

BXEventLog Table	423
BXEventLogDetail Table	424
BXEventSchedule Table	424
BXActivity Table	425
Document Tables	426
BxBillerCustomerAccount Table	426
BXContainer Table	428
BXDocument Table	429
BXSummary Table	430
Payment Related Tables	433
BXPayHistory Table	433
BXPaySchedule Table	435
Assignment Tables	436
BXUserAcctAsgt Table	436
BXDocLineAsgt Table	437
Other Tables	437
BXLoadTrack Table	438
BXMessage Table	438
BXExtensionDesc Table	439
BXError Table	440
 Chapter 6 Membership Schema	 441
Membership Schema	441
Relationship Between Tables	442
Listing of Structures	443
Biller Tables	443
nsBillerProfile Table	443
nsAccountType Table	445
nsGroup Table	446
nsPaymentType Table	446
Customer Tables	447
nsCustomerProfile Table	447
nsFinancialAccount Table	449
nsAddress Table	450
 Chapter 7 Payment Tables	 451
Payment Schema	451
Payment Information Table	451
BXCardInfo	452
 Chapter 8 Currency Conversion Tables	 453
Currency Conversion Schema	453

Listing of Tables	454
Currency Conversion Tables	454
Currency Table	454
CurrencyLoadTrack Table	455
CurrencyConversion	455
Appendix A Enumerated Domains	457
Domain Checking	457
The enumdef.conf File	458
Index	463

List of Tables

Table 1-1	Base and Core Classes	35
Table 1-2	ACGUIDException Class Definition	38
Table 1-3	ACGUIDGeneration Class Definition	39
Table 1-4	ACGUIDGeneration Class Methods	39
Table 1-5	IActivityLogger class definition	41
Table 1-6	IActivityLogger Class Methods	41
Table 1-7	ActivityLoggerAL class definition	42
Table 1-8	ActivityLoggerAL Class Methods	42
Table 1-9	ActivityLoggerMgr Class Definition	43
Table 1-10	ActivityLoggerMgr Class Methods	43
Table 1-11	AuthToken Class Definition	44
Table 1-12	AuthToken Class Methods	44
Table 1-13	AuthToken Class Fields	51
Table 1-14	BaseAppLogic Class Definition	52
Table 1-15	BaseAppLogic Class Field	52
Table 1-16	BXActivities Class Definition	53
Table 1-17	ActivityLoggerAL Class Methods	53
Table 1-18	BXActivity Class Definition	54
Table 1-19	BXActivity Class Methods	54
Table 1-20	BXActivity Class XML Constants	56
Table 1-21	BXActivity Class Activity Constants	57
Table 1-22	BXActivity Class Action Constants	58
Table 1-23	BXActivityCheck Class Definition	59
Table 1-24	BXActivityCheck Class Methods	60
Table 1-25	BXBaseDef Class Definition	61
Table 1-26	BXBaseDef Class Methods	61
Table 1-27	BXConfig Class Definition	62

Table 1-28	BXConfig Class Methods	62
Table 1-29	BXConverter Class Definition	66
Table 1-30	BXConverter Class Methods	66
Table 1-31	BXDatetime Class Definition	67
Table 1-32	BXDatetime Class Methods	67
Table 1-33	IBXDBContext Class Definition	72
Table 1-34	IBXDBContext Class Methods	73
Table 1-35	BXEntityObject Class Definition	75
Table 1-36	BXEntityObject Class Fields	75
Table 1-37	BXEntityObject Class Methods	76
Table 1-38	BXEnumDef Class Definition	79
Table 1-39	BXEnumDef Class Methods	79
Table 1-40	BXError Class Definition	81
Table 1-41	BXError Class Fields	81
Table 1-42	BXException Class Definition	82
Table 1-43	BXFileId Class Definition	82
Table 1-44	BXFileId Class Methods	83
Table 1-45	BXFilter Class Definition	85
Table 1-46	BXFilter Class Methods	85
Table 1-47	BXGlobal Class Definition	92
Table 1-48	BXGlobal Class Methods	92
Table 1-49	BXGlobal Class Fields	94
Table 1-50	BXJDBCHelper Class Definition	95
Table 1-51	BXJDBCHelper Class Methods	95
Table 1-52	BXMessages Class Definition	105
Table 1-53	BXMessages Class Methods	105
Table 1-54	BXObjectMgr Class Definition	106
Table 1-55	BXObjectMgr Class Method	106
Table 1-56	BXObjectMap Class Definition	107
Table 1-57	BXObjectMap Class Methods	107
Table 1-58	BXOrder Class Definition	109
Table 1-59	BXOrder Class Methods	109
Table 1-60	BXStatus Class Definition	113
Table 1-61	BXStatus Class Result Code Constants	114
Table 1-62	BXStatus Class Name Constants	114
Table 1-63	BXStatus Class Miscellaneous Constants	115
Table 1-64	BXStatus Class Methods	116

Table 1-65	BXStopwatch Class Definition	121
Table 1-66	BXStopwatch Class Methods	121
Table 1-67	BXTableDef Class Definition	124
Table 1-68	BXTableDef Class Methods	124
Table 1-69	IBXEntity Class Definition	128
Table 1-70	IBXEntity Class Methods	128
Table 1-71	SearchResultSet Class Definition	134
Table 1-72	SearchResultSet Fields	135
Table 2-1	Billing Classes	137
Table 2-2	IBXBillerCustAcct Interface Definition	139
Table 2-3	IBXBillerCustAcct Methods	140
Table 2-4	BXBillingContext Class Definition	162
Table 2-5	BXBillingContext Constructors	163
Table 2-6	BXBillingEntity Interface Definition	164
Table 2-7	IBXBillingEntity Interface Methods	165
Table 2-8	BXBillingMgr Class Definition	168
Table 2-9	BXBillingMgr Method Definitions	168
Table 2-10	IBXCardInfo Class Definition	172
Table 2-11	IBXCardInfo Methods	172
Table 2-12	IBXContainter Class Definition	185
Table 2-13	IBXContainer Class Methods	185
Table 2-14	IBXDetail Class Definition	194
Table 2-15	IBXDetail Class Methods	194
Table 2-16	BXDocument Class Definition	198
Table 2-17	IBXDocument Class Methods	198
Table 2-18	IBXEventLog Class Definition	216
Table 2-19	IBXEventLog Methods	216
Table 2-20	BXEventLogDetail Class Description	221
Table 2-21	BXEventLogDetail Class Methods	221
Table 2-22	IBXExtensionDesc Class Description	223
Table 2-23	IBXExtensionDesc Class Methods	224
Table 2-24	IBXLineItem Class Definition	226
Table 2-25	IBXLineItem Class Methods	226
Table 2-26	IBXMessage Class Definition	240
Table 2-27	IBXMessage Class Methods	240
Table 2-28	IBXOFXSignOn Class Definition	252
Table 2-29	IBXOFXSignOn Class Methods	252

Table 2-30	IBXPayHistory Class Definition	254
Table 2-31	IBXPayHistory Class Methods	254
Table 2-32	IBXPaySchedule Class Definition	268
Table 2-33	IBXPaySchedule Class Methods	268
Table 2-34	IBXSummary Class Definition	282
Table 2-35	IBXSummary Class Methods	282
Table 3-1	Membership Classes / Interfaces	306
Table 3-2	IBXAddress Interface Definition	307
Table 3-3	IBXAddress Class Methods	307
Table 3-4	IBXBillerProfile Interface Definition	313
Table 3-5	IBXBillerProfile Class Methods	313
Table 3-6	IBXCustomerProfile Interface Definition	337
Table 3-7	IBXCustomerProfile Interface Methods	337
Table 3-8	IBXFinancialAccount Interface Definition	363
Table 3-9	IBXFinancialAccount Interface Methods	363
Table 3-10	BXMembershipContext Class Definition	367
Table 3-11	BXMembershipContext Methods	368
Table 3-12	IBXMembershipEntity Interface Definition	372
Table 3-13	IBXMembershipEntity Interface Methods	372
Table 3-14	BXMembershipMgr Class Definition	373
Table 3-15	BXMembershipMgr Class Methods	373
Table 3-16	BXMembershipMgr Class Definition	377
Table 3-17	IBXPaymentType Interface Methods	378
Table 4-1	Event Classes	383
Table 4-2	IBXEvent Class Definition	384
Table 4-3	IBXEvent Class Methods	385
Table 4-4	BXEventContext Class Description	389
Table 4-5	IBXEventEntity Interface Definition	391
Table 4-6	IBXEventEntity Class Methods	391
Table 4-7	IBXEventHandler Class Description	392
Table 4-8	IBXEventHandler Class Methods	393
Table 4-9	BXEventManager Class Definition	399
Table 4-10	BXEventManager Class Methods	399
Table 4-11	IBXEventQueue Interface Definition	401
Table 4-12	IBXEventQueue Methods	402
Table 4-13	IBXHandler Class Definition	410
Table 4-14	IBXHandler Class Methods	411

Table 5-1	Oracle Database Billing Tables	419
Table 5-2	BXEvent Table	420
Table 5-3	BXHandler Table	420
Table 5-4	BXEventHandler Table	421
Table 5-5	BXEventQueue Table	422
Table 5-6	BXEventLog Table	423
Table 5-7	BXEventLogDetail Table	424
Table 5-8	BXEventSchedule Table	424
Table 5-9	BXActivity Table	425
Table 5-10	BXBillerCustomerAccount Table	426
Table 5-11	BXContainer Table	428
Table 5-12	BXDocument Table	429
Table 5-13	BXSummary Table	430
Table 5-14	BXPayHistory Table	433
Table 5-15	BXPaySchedule Table	435
Table 5-16	BXUserAcctAsgt Table	436
Table 5-17	BXDocLineAsgt Table	437
Table 5-18	BXLoadTrack Table	438
Table 5-19	BXMessage Table	438
Table 5-20	BXExtensionDesc Table	439
Table 5-21	BXError Table	440
Table 6-1	Membership Tables	443
Table 6-2	nsBillerProfile Table	443
Table 6-3	nsAccountType Table	445
Table 6-4	nsGroup Table	446
Table 6-5	nsPayment Table	446
Table 6-6	nsCustomerProfile Table	447
Table 6-7	nsFinancialAccount Table	449
Table 6-8	naAddress Table	450
Table 7-1	BXCardInfo Table	452
Table 8-1	Currency Tables	454
Table 8-2	Currency Table	454
Table 8-3	CurrencyLoadTrack Table	455
Table 8-4	CurrencyConversion Table	455
Table A-1	enumdef.conf File	458

About This Document

The BillerXpert *API & Schema Reference Manual* describes the interfaces to component business objects that you can use to customize BillerXpert B2B Edition. It also describes the database schemas for BillerXpert B2B Edition.

This preface contains the following sections:

- Audience
- What's in This Document
- Documentation Conventions
- The Document Online
- Related Documentation
- Product Support

Audience

The audience for this document is web site developers and Java programmers who want to modify the appearance of BillerXpert or its operation. You must have experience with HTML and Java and have access to the iPlanet Application Server (iAS) documentation:

`http://docs.iplanet.com/docs/manuals/ias.html`

You need to use this manual if you are changing or extending BillerXpert, changing the database schema, or adding event handlers.

Required Documents

When customizing BillerXpert, you may need to refer to the *iPlanet BillerXpert Customization Guide* for information about and examples of using component business object interfaces and event handlers.

You will also need to have read or have available the following iPlanet Application Server (iAS) documents:

- *iAS Java Programmer's Guide* — Provides background information about using the iAS programming constructs upon which BillerXpert was developed.
(<http://docs.iplanet.com/docs/manuals/ias/60/sp2/JavaProgGuide/contents.htm>.)
- *IAS Foundation Class Reference (Java)* — Provides descriptions of classes that BillerXpert uses or classes from which BillerXpert classes are derived.
(<http://docs.iplanet.com/docs/manuals/ias/60/sp2/javaapi/contents.htm>.)

What's in This Document

The following table summarizes what each chapter covers.

Table 1 Chapter Summary

If you want to know about this	See this chapter
Describes the contents of this guide; listing of documentation set; information on product support.	“About This Document”
Introduction to the component business object interface classes and database schemas for BillerXpert	“Introduction to BillerXpert B2B Classes and Database Schemas”
Lists the following BuyerXpert Java classes and methods: • Base and core classes • Billing • Membership • Event handlers	Part 1, “BillerXpert B2B Classes”
Describes the interfaces to the base and core classes, which provide basic functions such as activity tracking or error handling, which is shared by all BillerXpert components.	Chapter 1, “Base and Component Business Object Interfaces”
Describes the component business object interface classes for billing records. These classes provide access to records stored in an Oracle database.	Chapter 2, “Billing Classes Component Business Object Interfaces”

Table 1 Chapter Summary

If you want to know about this	See this chapter
Describes the component business object interface classes for membership records. These classes provide access to records stored in a Directory Server database using the Lightweight Directory Access Protocol (LDAP).	Chapter 3, “Membership Component Business Object Interfaces”
Describes the component business object interface classes for events. These classes provide events and event handlers for handling asynchronous events within BillerXpert.	Chapter 4, “Event Component Business Object Interfaces”
Lists the following BuyerXpert schemas: • Membership schema • Billing schema • Payment tables • Currency conversion tables	Part 2, “BillerXpert Database Schemas”
Describes the structure of the Oracle database used by BillerXpert	Chapter 5, “Billing Schema”
Describes the structure of the Directory Server database used by BillerXpert.	Chapter 6, “Membership Schema”
Describes the structure of the payment tables used by BillerXpert.	Chapter 7, “Payment Tables”
Describes the structure of the currency conversion tables used by BillerXpert.	Chapter 8, “Currency Conversion Tables”
Describes the domains that an implementation of a component business object interface checks before performing actions against a data repository.	Appendix A, “Enumerated Domains”

Documentation Conventions

This document uses the following conventions:

- The `monospace` font is used for sample code and code listings, Application Program Interface (API) and language elements (such as method names and property names), file names, path names, directory names, Hypertext Markup Language (HTML) tags, and any text that must be typed on the screen.
- The *italic* font is used in code to represent placeholder parameters (variables) that should be replaced with an actual value.

- Brackets ([]) are used to enclose optional parameters.
- A slash (/) is used to separate directories in a path. (Windows NT supports both the slash and the backslash.)

The Document Online

For an overview of BillerXpert 4.5 changes, refer to the BillerXpert 4.5 Release Notes.

An electronic version of these notes are available at:

<http://docs.iplanet.com/docs/manuals/>

Related Documentation

The BillerXpert documentation set includes:

- *Release Notes*—Contains important information on the current release of BillerXpert. Read this document before working with the new BillerXpert release.
- *Administrator's Guide*—Provides reference information and instructions on administering a fully-installed BillerXpert system.
- *Installation Guide*—Provides instructions for installing the BillerXpert product and its enabling software.
- *Customization Guide*—Provides guidelines and instructions for customizing the BillerXpert system.

Documentation for all iPlanet products can be found at the following web site:

<http://docs.iplanet.com/docs/manuals/>

Product Support

If you have problems with your BillerXpert system, contact iPlanet customer support using one of the following mechanisms:

- iPlanet online support web site at:

<http://www.iplanet.com/support/online/>

From this location, the CaseTracker and CaseView tools are available for logging problems.

- The telephone dispatch number associated with your maintenance contract

So that the technical support staff can best assist you in resolving problems, please have the following information available when you contact support:

- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps

Introduction to BillerXpert B2B Classes and Database Schemas

This chapter introduces the component business object interface classes and database schemas. For information about BillerXpert application-related classes, see the *iPlanet BillerXpert B2B Edition Customization Guide*.

This chapter presents the following topics:

- Component Business Object Interface Classes
- Database Tables

Component Business Object Interface Classes

BillerXpert B2B Edition contains component business objects that are Java interface based. There are components:

- Membership
- Billing

These component objects manage data storage, such as creation, storage, and retrieval of records, or contain processing logic.

Component Objects

Each component has its own factory class, which is used to create the object, and its own context class. A component object must be associated with a context before it can perform data access or processing. The responsibilities of these contexts differ depending on the kind of component.

Table 2 lists these components:

Table 2 Component Objects

Component	Context Responsibilities
Membership	None.; however, the context is required by convention.
Billing	Establishes a context between the component and the component object; responsible for database connections and transaction management.

Base and Core Classes

Component business object classes are derived from the IBXEntity class. In addition, BillerXpert provides several core classes that you can use with component business objects.

Table 3 shows the base classes for component business object interfaces and their contexts. These classes are located in Chapter 1, “Base and Component Business Object Interfaces.

Table 3 Base and Core Classes *(1 of 2)*

Class Name	Description
1. ACGUIDException	Throws an exception if there is a problem generating a unique ID.
2. ACGUIDGeneration	Represents a unique ID generator.
3. ActivityLogger	Base class for logging activity messages in the Oracle database.
4. IActivityLogger	Interface class for the activity logger to log activity messates in the Oracle database.
5. ActivityLoggerAL	AppLogic to log activity messages.
6. ActivityLoggerMgr	Creates an instance of the activity logger object.
7. AuthToken	Models an authentication token that can be used to verify credentials by a client.
8. BaseAppLogic	Base AppLogic that is reused by many modules.
9. BXActivities	Creates a Resource Bundle used for internationalization to lookup messages properties files.
10. BXActivity	Represents an activity to be logged.
11. BXActivityCheck	Checks if the activity is proper. Used by BXActivities.
12. BXBaseDef	Base class for name, type, value (NTV) lists.

Table 3 Base and Core Classes (2 of 2)

Class Name	Description
13. BXConfig	Represents configuration information.
14. BXConverter	Converts an NTV list to a ValList and vice versa.
15. BXDatetime	Converts a Java date to an iAS date and vice versa.
16. BXDBContext	Base class for component business object contexts.
17. IBXDBContext	Interface class for component business object contexts.
18. BXDBDef	Holds all the enumerations in the base class.
19. BXDef	Contains the class definitions for all enumerated types.
20. BXEntityObject	Base class for the entity bean.
21. BXEnumDef	Retrieves enumerator types. Reads from <code>enumdef.conf</code> .
22. BXError	Represents an error. Implements <code>IBXErrorEntity</code> . Stores entity bean errors.
23. BXException	Base exception class. All other exceptions should extend from this.
24. BXFileId	Retrieves the next FILEID in a sequence.
25. BXFilter	Represents a search filter.
26. BXGlobal	Keeps an NTV list for each biller's table definition.
27. BXJDBCHelper	Helper class that performs JDBC inserts, deletes, updates at the EJB level.
28. BXMessages	Reads the messages and the locale information from <code>billx2b.conf</code> .
29. BXObjectMgr	Resolves mapping between Interface Name and the actual Class Implementation depending upon the billerName and billType.
30. BXObjMap	Used by <code>BXObjectMgr</code> . Resolves mapping between the interface name and the actual class implementation, depending on the billName and the billType.
31. BXOrder	Represents the order in which search results are returned.
32. BXStatus	Represents a stack of status information.
33. BXStopwatch	Represents the elapsed time, in minutes and seconds, as a string. Used for performance testing and the EJB level.
34. BXTableDef	Performs a lookup to match the Oracle table and column names to fields stored in LDAP.
35. IBXEntity	Interface class for component business objects.
36. SearchResultSet	Returns the search result set. Used for JDBC lookups.

Billing Classes

Billing classes represent billing records in the database. You use these classes to manipulate billing objects.

Table 4 shows the billing classes and interfaces for component business object interfaces and their contexts. These classes are located in Chapter 2, “Billing Classes Component Business Object Interfaces.

Table 4 Billing Classes (1 of 2)

Class Name	Description
1. BXBillerCustAcct	Base class that manipulates BillerXpert customer account objects.
2. IBXBillerCustAcct	Interface that manipulates BillerXpert customer account objects.
3. BXBillingContext	Communicate between an AppLogic and an EJB component.
4. BXBillingEntity	Base class for billing business objects.
5. IBXBillingEntity	Interface for billing business objects.
6. BXBillingMgr	Creates billing-related component business objects.
7. BXCardInfo	Base class for credit card information.
8. IBXCardInfo	Interface for credit card information.
9. BXContainer	Base class that manipulates container objects.
10. IBXContainer	Interface that manipulates container objects.
11. BXDetail	Base class that manipulates detail line objects
12. IBXDetail	Interface that manipulates detail line objects
13. BXDocument	Base class that manipulates billing objects.
14. IBXDocument	Interface that manipulates billing objects.
15. BXEventLog	Base class that manipulates event log objects.
16. IBXEventLog	Interface that manipulates event log objects.
17. BXEventLogDetail	Manipulates event log detail objects.
18. BXExtensionDesc	Base class that represents the extension description table.
19. IBXExtensionDesc	Interface that represents the extension description table.
20. BXLineItem	Base class that manipulates objects that relate to purchase order line items.
21. IBXLineItem	Interface that manipulates objects that relate to purchase order line items.
22. BXMessage	Base class that manipulates message objects.

Table 4 Billing Classes (2 of 2)

Class Name	Description
23. IBXMessage	Interface that manipulates message objects.
24. BXPayHistory	Manipulates payment history objects.
25. IBXPayHistory	Manipulates payment schedule objects.
26. BXPaySchedule	Base class that manipulates payment schedule objects.
27. IBXPaySchedule	Interface that manipulates payment schedule objects.
28. BXSummary	Base class that manipulates billing summary objects.
29. IBXSummary	Interface that manipulates billing summary objects.

Membership Classes

Membership classes represent LDAP records on the Directory Server. You use these classes to manipulate membership objects.

Table 5 shows the membership classes and interfaces for component business object interfaces and their contexts. These classes are located in Chapter 3, “Membership Component Business Object Interfaces.

Table 5 Membership Classes / Interfaces (1 of 2)

Class Name	Description
1. BXAddress	Implementation of IBXAddress.
2. IBXAddress	Interface that defines address objects.
3. BXBillerProfile	Implementation of IBXBillerProfile.
4. IBXBillerProfile	Interface that defines billing organization objects.
5. BXCumulativeProfile	Implementation of IBXCumulativeProfile.
6. IBXCumulativeProfile	Interface that defines customer profile objects.
7. BXFinancialAccount	Implementation of IBXFinancialAccount.
8. IBXFinancialAccount	Manipulates financial account objects.
9. BXMembershipContext	Establishes a context between an AppLogic and a membership business object.
10. BXMembershipEntity	Base class that is subclasses by all membership classes.
11. IBXMembershipEntity	Base interface for membership business objects.
12. BXMembershipMgr	Creates membership-related business objects.

Table 5 Membership Classes / Interfaces (2 of 2)

Class Name	Description
13. BXPaymentType	Implementation of IBXPaymentType.
14. IBXPaymentType	Interface that defines payment type objects.

Event Classes

Event classes represent event records in the database. The event server uses these classes to manipulate event-related objects.

Table 6 shows the event classes and interfaces for component business object interfaces and their contexts. These classes are located in Chapter 4, “Event Component Business Object Interfaces.

Table 6 Event Classes

Class Name	Description
1. BXEvent	Base class that manipulates event objects.
2. IBXEvent	Interface class that manipulates event objects.
3. BXEventContext	Establishes a context between an AppLogic and a billing component business object.
4. BXEventEntity	Base class that manipulates event-related objects.
5. IBXEventEntity	Interface class that manipulates event-related objects.
6. BXEventHandler	Base class that manipulates event handler mapping objects.
7. IBXEventHandler	Interface class that manipulates event handler mapping objects.
8. BXEventManager	Creates event-related component business objects.
9. BXEventQueue	Base class that manipulates event queue objects.
10. IBXEventQueue	Interface class that manipulates event queue objects.
11. BXHandler	Base class that manipulates handler objects.
12. IBXHandler	Interface class that manipulates handler objects.

Database Tables

BillerXpert B2B Edition contains the following database tables:

- Billing Schema

- Member Schema
- Payment Tables
- Currency Conversion Tables

Table 7 identifies the tables in the Oracle database for BillerXpert.

Table 7 Database Tables

Schema	Description
1. BXActivity	Activity Log
2. BXBillerCustomerAccount	BillerXpert customer account
3. BXContainer	Containers
4. BXDocLineAsgt	Relationship between documents and detail lines
5. BXDocument	Documents
6. BXError	Error Logs
7. BXEvent	Events
8. BXEventHandler	Event handlers
9. BXEventLog	Event logs
10. BXEventLogDetail	Event log detail lines
11. BXEventQueue	Event queue entries
12. BXEventSchedule	Event Scheduling
13. BXExtensionDesc	Extension Description
14. BXHandler	Handlers
15. BXLineItemTelCoAll	Call details
16. BXLineItemTelCoMisc	Miscellaneous details
17. BXLineItemTelCoRating	Rating details
18. BXLloadTrack	Loader tracking
19. BXMessage	Messages
20. BXPayHistory	Payment histories
21. BXPaySchedule	Payment schedules
22. BXSummary	Summary lines
23. BXUserAcctAsgt	Relationship between users and accounts
24. BXCardInfo	Stores credit card related information

LDAP Schema

Table 8 identifies the LDAP schema for BillerXpert.

Table 8 LDAP Schema

Schema	Description
1. nsAddressTable	Address object. Stores addresses of people.
2. nsBillerProfile	Stores biller information.
3. nsCompanyProfile	Stores company information.
4. nsCustomerProfile	Stores membership information for BSP, Biller, and company.
5. nsDynamicGroup	Stores information about ojects that use dynamic groups. These are used for storing roles, privileges, and payment rules.
6. nsFinancialAccount	Stores information about the financial account.Used for both Biller receiving accounts and company’s payment accounts.
7. nsPaymentType	Stores customer payment information.
8. nsPaymentVender	Stores information about payment vendors.

Enumerated Domains

Appendix A, “Enumerated Domains” discusses the enumerated domains checking and the `enumdef.conf` configuration file.

BillerXpert B2B Classes

Chapter 1, “Base and Component Business Object Interfaces”

Chapter 2, “Billing Classes Component Business Object Interfaces”

Chapter 3, “Membership Component Business Object Interfaces”

Chapter 4, “Event Component Business Object Interfaces”

Base and Component Business Object Interfaces

This chapter discusses interfaces to base and core classes. These classes are the base classes for component business object interfaces or classes that provide a basic function, such as activity tracking or error handling, which is shared by all components in BillerXpert.

This chapter includes the following sections:

- Overview
- Base and Core Class Reference Descriptions

Overview

BillerXpert provides a class from which database records and contexts are derived. BillerXpert also provides several classes you can use to specify search criteria, as well as classes for activity tracking, manipulating configuration files, checking status information, and for generating unique IDs for database records. Table 1-1 lists these classes.

Table 1-1 Base and Core Classes *(1 of 3)*

Class Name	Description	Location
1. ACGUIDException	Throws an exception if there is a problem generating a unique ID.	page 38
2. ACGUIDGeneration	Represents a unique ID generator.	page 38
3. ActivityLogger	Base class for logging activity messages in the Oracle database.	page 40

Table 1-1 Base and Core Classes (2 of 3)

Class Name	Description	Location
4. IActivityLogger	Interface class for the activity logger to log activity messages in the Oracle database.	page 40
5. ActivityLoggerAL	AppLogic to log activity messages.	page 41
6. ActivityLoggerMgr	Creates an instance of the activity logger object.	page 43
7. AuthToken	Models an authentication token that can be used to verify credentials by a client.	page 43
8. BaseAppLogic	Base AppLogic that is reused by many modules.	page 51
9. BXActivities	Creates a Resource Bundle used for internationalization to lookup messages properties files.	page 53
10. BXActivity	Represents an activity to be logged.	page 54
11. BXActivityCheck	Checks if the activity is proper. Used by BXActivity.	page 59
12. BXBaseDef	Base class for name, type, value (NTV) lists.	page 61
13. BXConfig	Represents configuration information.	page 62
14. BXConverter	Converts an NTV list to a ValList and vice versa.	page 65
15. BXDatetime	Converts a Java date to an iAS date and vice versa.	page 67
16. BXDBContext	Base class for component business object contexts.	page 72
17. IBXDBContext	Interface class for component business object contexts.	page 72
18. BXDBDef	Holds all the enumerations in the base class.	page 75
19. BXDef	Contains the class definitions for all enumerated types.	page 75
20. BXEntityObject	Base class for the entity bean.	page 75
21. BXEnumDef	Retrieves enumerator types. Reads from <code>enumdef.conf</code> .	page 79
22. BXError	Represents an error. Implements IBXErrorEntity. Stores entity bean errors.	page 81
23. BXException	Base exception class. All other exceptions should extend from this.	page 82
24. BXFileId	Retrieves the next FILEID in a sequence.	page 82
25. BXFilter	Represents a search filter.	page 85

Table 1-1 Base and Core Classes (3 of 3)

Class Name	Description	Location
26. BXGlobal	Keeps an NTV list for each biller's table definition.	page 85
27. BXJDBCHelper	Helper class that performs JDBC inserts, deletes, updates at the EJB level.	page 95
28. BXMessages	Reads the messages from <code>billx2b.conf</code> .	page 104
29. BXObjectMgr	Resolves mapping between InterFace Name and the actual Class Implementation depending upon the billerName and billType.	page 106
30. BXObjMap	Used by BXObjectMgr. Resolves mapping between the interface name and the actual class implementation, depending on the billName and the billType.	page 107
31. BXOrder	Represents the order in which search results are returned.	page 109
32. BXStatus	Represents a stack of status information.	page 113
33. BXStopwatch	Represents the elapsed time, in minutes and seconds, as a string. Used for performance testing at the EJB level.	page 121
34. BXTableDef	Performs a lookup to match the Oracle table and column names to fields stored in LDAP.	page 124
35. IBXEntity	Interface class for component business objects.	page 127
36. SearchResultSet	Returns the search result set. Used for JDBC lookups.	page 134

ACGUIDException Class Reference

The ACGUIDException class throws an exception if there is a problem generating a unique ID.

Class Definition

Table 1-2 describes the ACGUIDException class definition.

Table 1-2 ACGUIDException Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.core.guid;</code>
Syntax	<code>public class ACGUIDException extends Exception {...}</code>

Constructor

The ACGUIDException class contains the following constructor:

Syntax	<code>public ACGUIDException() public ACGUIDException (String s)</code>
Parameters	<code>s</code> — A string that specifies the <code>s</code> object.
Discussion	Creates an ACGUIDException object.

ACGUIDGeneration Class Reference

The ACGUIDGeneration class represent unique IDs that identify records in a database. These IDs consist of the class name of the corresponding object, the internet address of the local host, and a timestamp.

Class Definition

Table 1-2 describes the ACGUIDGeneration class definition.

Table 1-3 ACGUIDGeneration Class Definition

Package	com.iplanet.ecomm.billxb2b.core.guid;
Syntax	public class ACGUIDGeneration {...}

Methods

Table 1-4 lists the ACGUIDGeneration class methods.

Table 1-4 ACGUIDGeneration Class Methods

Method	Description
CreateGUID()	Creates a unique ID.
finalize()	Releases the resources when the object is no longer referred to and is about to be collected by the garbage collector.
GetObjectType()	Determines the class name.
Init()	Sets up host information in the object.

CreateGUID()

Creates a unique ID.

Contents	Description
Syntax	public final String CreateGUID(String <i>objType</i>) throws ACGUIDException;
	public final String [] CreateGUID (String <i>objType</i> , int <i>nCount</i>) throws ACGUIDException;
Parameters	<i>objType</i> — The object type for this unique ID. It is appended to the end of the unique ID string.
	<i>nCount</i> — The number of unique IDs to create and return in the string [].
Discussion	

finalize()

Syntax	<code>protected final void finalize() throws java.lang.Throwable;</code>
Parameters	None.
Discussion	

GetObjectType()

Determines the class name.

Syntax	<code>public final String GetObjectType (String <i>guid</i>) throws ACGUIDException;</code>
Parameters	<i>guid</i> — A string that contains a unique ID
Returns	A string that contains the class name from the unique ID.

Init()

Sets up host information in the object.

Syntax	<code>public void Init() throws java.net.UnknownHostException;</code>
Parameters	None.
Discussion	You must call the Init() method before calling the CreateGUID() method. See the example for the CreateGUID() method in the ACGUIDGeneration Class Reference.

ActivityLogger Class Reference

The IActivityLogger class is the base class for logging activity messages in the Oracle database. Whenever possible, use the interface class IActivityLogger (see “IBXEntity Class Reference” on page 127) instead of ActivityLogger.

IActivityLogger Class Reference

The IActivityLogger class is the interface class for logging activity messages in the Oracle database.

Class Definition

Table 1-7 describes the IActivityLogger class definition.

Table 1-5 IActivityLogger class definition

Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public interface IActivityLogger {...}</code>

Methods

Table 1-12 lists the IActivityLogger class method.

Table 1-6 IActivityLogger Class Methods

Method	Description
<code>process()</code>	Processes activity and inserts into BXActivity table.

process()

Processes activity and inserts the information into the BXActivity table.

Syntax	<code>boolean process(iContext context, IList value);</code>
Parameters	<i>context</i> — iContext object <i>value</i> — IList object that specifies the contents of a log entry
Discussion	

ActivityLoggerAL Class Reference

The ActivityLoggerAL class uses AppLogic to log activity messages. It uses the IActivityLogger class to insert entries into the BXActivity table.

Class Definition

Table 1-7 describes the ActivityLoggerAL class definition.

Table 1-7 ActivityLoggerAL class definition

Package	com.iplanet.ecomm.billxb2b.core;
Syntax	public class ActivityLoggerAL extends BaseAppLogic {...}

Methods

This section describes the ActivityLoggerAL class methods. Table 1-12 lists these methods.

Table 1-8 ActivityLoggerAL Class Methods

Method	Description
execute()	
guid()	

guid()

Syntax	public java.lang.String guid();
Parameters	None.
Discussion	Overrides guid() in the AppLogic class.

execute()

Syntax	public int execute();
Parameters	None.
Discussion	Overrides execute() in the AppLogic class.

ActivityLoggerMgr Class Reference

The ActivityLoggerMgr class creates an instance of the activity logger object.

Class Definition

Table 1-9 describes the ActivityLoggerMgr class definition.

Table 1-9	ActivityLoggerMgr Class Definition
Package	com.iplanet.ecomm.billxb2b.core;
Syntax	public ActivityLoggerMgr {...}

Methods

This section describes the ActivityLoggerMgr class methods. Table 1-10 lists these methods.

Table 1-10 ActivityLoggerMgr Class Methods

Method	Description
createActivityServer()	Creates an instance by Name.

createActivityServer()

Syntax	public static IactivityLogger createActivityServer() throws java.lang.Exception;
Parameters	None.
Discussion	

AuthToken Class Reference

The AuthToken class models an authentication token that can be used to verify previously submitted credentials by a client.

Class Definition

Table 1-11 describes the AuthToken class definition.

Table 1-11 AuthToken Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class AuthToken extends java.lang.Object {...}</code>

Methods

This section describes the AuthToken class methods. Table 1-12 lists these methods.

Table 1-12 AuthToken Class Methods

Method	Description
<code>create()</code>	Creates an authorization token with the given name.
<code>createFromToken()</code>	Creates a From authorization token.
<code>get()</code>	Retrieves the key for an authorization token.
<code>getExpiration()</code>	Retrieves the expiration time and date for an authorization token.
<code>getName()</code>	Returns the expiration time and date for the token.
<code>hasExpired()</code>	Retrieves the authorization token name
<code>isValid()</code>	Returns true if the token is valid (has not expired) and the recomputed signatures match the previous results of <code>sign()</code> .
<code>put()</code>	Puts a value in the authorization token under the key specified.
<code>setAbsoluteExpiration()</code>	Sets the internal token expiration to the absolute specified.
<code>setExpiration()</code>	Sets the internal token expiration to <code>deltaSeconds</code> from the current time.
<code>setInternalSecret()</code>	
<code>setName()</code>	Returns a value in the authorization token under the key specified. If key is null, null is returned.
<code>sign()</code>	Signs an authorization token.

Table 1-12 AuthToken Class Methods

Method	Description
<code>signData()</code>	Signs an authorization token so that it can be used.
<code>toHex()</code>	Converts an ASCII string into a hexadecimal format.
<code>toHttpCookie()</code>	Converts the token into an HTTP Right Hand Side of the set-Cookie header.
<code>toParameters()</code>	Converts the token to a list of attribute value pairs in a String format
<code>toString()</code>	Returns a string representation of an authorization token.
<code>toToken()</code>	Return the token encoding of the AuthToken object. Returns the token as a Hex encoded string.

create()

Creates an authorization token with the given name.

	<code>public static AuthToken create();</code>
Syntax	<code>public static AuthToken create (String <i>name</i>);</code> <code>public static AuthToken create (String <i>name</i>, String <i>md</i>);</code>
Parameters	<i>name</i> — A string that specifies the authorization token name <i>md</i> — A string that specifies the message digest algorithm
Discussion	Creates an authorization token and sets the message digest algorithm value.

createFromToken()

Creates an authorization token.

Syntax	<code>public static AuthToken createFromToken (String <i>token</i>);</code>
Parameters	<i>token</i> — A string that specifies the authorization token name
Discussion	Creates a new authorization token with the specified name.

get()

Retrieves the key for the authorization token.

Syntax	<code>public boolean get (String key) ;</code>
Parameters	<i>key</i> — A long that specifies the specified key for the authorization token.
Discussion	Retrieves the key for the authorization token. Returns the value set or null.

getExpiration()

Retrieves the expiration time and date for the authorization token.

Syntax	<code>public Date getExpiration() ;</code>
Parameters	None.
Discussion	Retrieves the internal token expiration date at the absolute time that is specified.

getName()

Retrieves the authorization token name.

Syntax	<code>public String getName() ;</code>
Parameters	None.
Discussion	Returns the name of the authorization token.

hasExpired()

Determines if an authorization token has expired.

Syntax	<code>public boolean hasExpired() ;</code>
Parameters	None.
Discussion	Checks to see if an authorization token has expired. Returns true if the token has expired and false if it has not expired.

isValid()

Determines if an authorization token is valid. A valid token has not expired and its recomputed signatures match the previous result of `sign()`.

Syntax	<code>public boolean isValid();</code>
Parameters	None.
Discussion	Returns true if the token has not expired or false otherwise.

put()

Puts a value in the authorization token under the key specified.

Syntax	<code>public String put (String key, String value);</code>
Parameters	<i>key</i> — A string that specifies the parameter key to use (for example, <code>id</code>) <i>value</i> — A long that specifies the value in the authorization token under the specified key. If <i>key</i> is null, returns null.
Discussion	Both <i>key</i> and <i>value</i> must not be null.

setAbsoluteExpiration()

Sets the internal token expiration time at the absolute time that is specified.

Syntax	<code>public void setAbsoluteExpiration (long expireAtSeconds);</code>
Parameters	<i>expireAtSeconds</i> — A long that specifies the absolute number of seconds until an authorization token expires. If you set the expiration to zero, the token will never expire.
Discussion	If you set the expiration time to zero, the token will never expire.

setExpiration()

Sets the token expiration time and date.

Syntax	<code>public void setExpiration(long deltaSeconds);</code>
Parameters	<i>deltaSeconds</i> — A long that specifies the amount of seconds for which the token is considered valid.

Discussion	Returns the expiration time and date for the token. <code>deltaSeconds</code> is the amount of seconds for which the token is valid. Calls <code>setAbsoluteExpiration()</code> .

setInternalSecret()

Sets the internal secret to use. The internal secret is a random series of bytes that introduce confusion to the final signature.

Syntax	<code>void setInternalSecret(byte b[]);</code>
Parameters	<i>b</i> — A byte that specifies the internal secret to be set. Can be any array of bytes > 0.
Discussion	If you do not set the internal secret, the default value stored in <code>gsSecret</code> is used.

setName()

Sets the name of the authorization token.

Syntax	<code>public void setName (String name);</code>
Parameters	<i>name</i> — A string that specifies the name of this token
Discussion	Sets the authorization token name to the values of <i>name</i> .

sign()

Signs an authorization token so it can be used.

Syntax	<code>public String sign() throws Exception;</code>
Parameters	None.
Discussion	Returns the signature in a string form that is typically Message Digest algorithm dependent.

signData()

Signs an authorization token so that it can be used. For example, returns a valid token value via `toToken()` or `toHttpCookie()`.

Syntax	<code>protected String signData (String data) throws Exception;</code>
Parameters	<i>data</i> — An optional string that specifies the data used in signing the authorization token.
Discussion	Returns the signature in a string format that is typically Message Digest dependent.

toHex()

Converts an ascii string into hexadecimal format.

Syntax	<code>static public String toHex(String str);</code>
Parameters	<i>str</i> — A string that specifies the string value
Discussion	Returns a string representation of the authorization token.

toHttpCookie()

Converts an authorization token into an HTTP Right Hand Side of the Set-Cookie header.

Syntax	<code>String toHttpCookie(boolean bExpires, String path, String domain, boolean bSecure);</code>
Parameters	<i>bExpires</i> — If true ; expires= is added to the header string
	<i>path</i> — If not null ; path= is added to the header string
	<i>domain</i> — If not null ; domain= is added to the header string
	<i>bSecure</i> — If not null ; bSecure= is added to the header string
Discussion	

toParameters()

Converts an authorization token to a list of attribute value pairs.

Syntax	<code>public String toParameters(boolean <i>bIncludeChecksum</i>) ;</code>
Parameters	<i>bIncludeChecksum</i> — Checksum of the authorization token. If set to true, the checksum of the token computed via <code>sign()</code> is appended to this list as <code>m="checksum_value"</code> .
Discussion	Returns the authorization token encoded as a list of parameters.

toString()

Returns a string representation of an authorization token.

Syntax	<code>public String toString();</code>
Parameters	None.
Discussion	

toToken()

Returns the token encoding of an authorization token as a hexadecimal encoded string.

Syntax	<code>public String toToken();</code>
Parameters	None.
Discussion	Converts the authorization token to a list of attribute value pairs in a string format. <code><key>="<val>" <key1>="<val1>" ...</code>

setInternalSecret()

Creates the token from a hex encoded string (such as the one produced in `toToken()`).

Syntax	<code>void setInternalSecret(byte <i>b[]</i>) ;</code>

Parameters	<i>b[]</i> — A byte that specifies the internal secret to be set.
Discussion	The token returned can be checked to see if it is valid (using <code>isValid()</code>) and if it has expired (using <code>hasExpired()</code>). If both conditions are true, the token is valid.

Fields

The `AuthToken` class contains the following fields that the server uses to create an authentication code.

Table 1-13 AuthToken Class Fields

Field	Description
<code>AUTH_TOKEN_NAME</code>	Name of the authorization token.
<code>USERID</code>	User Id.
<code>TYPE</code>	Not currently used.
<code>DN</code>	Not currently used.
<code>KEY1</code>	Used to create an authorization token with a specified key.
<code>KEY2</code>	
<code>KEY3</code>	
<code>mlExpiration</code>	Absolute expiration date of the token (zero). The token never expires when this field is set.
<code>moKey</code>	Key vector (parallel vector to <code>moData</code>).
<code>moData</code>	Value vector (parallel vector to <code>moKey</code>).
<code>msChecksum</code>	Checksum computed for the token using <code>sign()</code> or <code>signdata()</code> .
<code>msName</code>	Name of the authorization token.
<code>msMD</code>	Default Message Digest algorithm.
<code>USERID</code>	Used to create an authorization token with a specified <code>USERID</code> .

BaseAppLogic Class Reference

The `BaseAppLogic` class contains the base `AppLogic` that is reused by many modules.

Class Definition

Table 1-9 describes the BaseAppLogic class definition.

Table 1-14 BaseAppLogic Class Definition

Package	com.iplanet.ecomm.billxb2b.core;
Syntax	public class BaseAppLogic extends AppLogic {...}

Constructor

The BaseAppLogic class contains the following constructor.

Syntax	public BaseAppLogic extends Applogic;
Parameters	None.
Discussion	Creates a BaseAppLogic object.

Fields

This section lists the BaseAppLogic class field.

Table 1-15 BaseAppLogic Class Field

Field	Description
public String appName	The name of your application, which should be the same for all AppLogic in your application.

Syntax	public String BaseAppLogic()
Discussion	Creates an instance by Name.

BXActivities Class Reference

The BXActivities class retrieves messages from the activity message file. The BXActivities class is used for internationalization, as well as localization. It creates a standard Java Resource Bundle used to retrieve localizable data and activity messages from a properties file.

The activities properties file it reads from it is based on the language and country code set in the first `index.html` page for a particular biller. For example, if the parameters of language “en” and country “US” are passed, the BXActivities class will read from the `Activities-en-US.properties` file located under your `$IAS_HOME/config` directory.

Class Definition

Table 1-16 describes the BXActivities class definition.

Table 1-16 BXActivities Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXActivities {...}</code>

Methods

This section describes the BXActivities class methods. Table 1-17 lists these methods.

Table 1-17 ActivityLoggerAL Class Methods

Method	Description
<code>getActivityMessage()</code>	Retrieves a message from the Activities properties file.
<code>init()</code>	Retrieves the file name from the configuration file.

`getActivityMessage()`

Retrieves a message from the activity message file.

Syntax	<code>public static java.lang.String getActivityMessage(String label);</code>
Parameters	<i>label</i> — A string that specifies the activity message label.

Discussion	
	init() Retrieves the file name from the configuration file.
Syntax	<pre>public static void init();</pre>
Parameters	None.
Discussion	

BXActivity Class Reference

The BXActivity class represents activity information written to the activity log file in XML format. You should only use the `LogActivity()` method from this class.

The default activity log file is set by the LDAP `ACTIVITY_FILENAME`.

BXActivity Class Definition

Table 1-18 describes the BXActivity class definition.

Table 1-18 BXActivity Class Definition	
Package	<code>com.iplanet.ecomm.billxb2b.core.util;</code>
Syntax	<code>public interface BXActivity {...}</code>

BXActivity Class Methods

This section describes the BXActivity class methods. Table 1-19 lists the BXActivity class methods.

Table 1-19 BXActivity Class Methods	
Method	Description
<code>LogActivity()</code>	Writes XML formatted data to the activity log file.
<code>xmlWrite()</code>	Returns the expiration time and date for the token.

Table 1-19 BXActivity Class Methods

Method	Description
<code>release()</code>	Sets the internal token expiration to the absolute time specified.
<code>init()</code>	

LogActivity()

`LogActivity()` writes XML-formatted data to the activity log file.

Syntax	<pre>public static void LogActivity(IContext context, IValList list, boolean check) throws Exception;</pre> <code>context</code> — An IContext object
Parameters	<code>list</code> — An IValList object that specifies the contents of the activity log entry. <code>check</code> — Not used, specify false.
Discussion	The <code>LogActivity()</code> method writes the specified list to the activity tracking file. The following method creates an activity log entry.
Example	<pre>public static void createCustomerInserted (IContext context, String alName, String userID, String customerID, String billerID) throws Exception { try { IValList data = GX.CreateValList(); data.setValString(BXActivity.APPLNAME, APPNAME); data.setValString(BXActivity.CREATEDBY, alName); data.setValString(BXActivity.SUBJECT, BXActivity.CUSTOMER); data.setValString(BXActivity.ACTION, BXActivity.INSERTED); data.setValString(BXActivity.ORGID, billerID); data.setValString("USER_ID", userID); data.setValString("CUSTOMER_ID", customerID); BXActivity.LogActivity(context, data, false); } catch (Exception e) { throw e; } }</pre>

xmlWrite()

Syntax	<code>public static void xmlWrite (String <i>buffer</i>) throws Exception;</code>
Parameters	<i>buffer</i> — A string that specifies the buffer for writing XML.
Discussion	

init()

Contents	Description
Syntax	<code>public static void int() throws Exception;</code>
Parameters	None.
Discussion	

XML Constants

The following fields specify the tags associated with the XML entry. Table 1-20 lists the activity constants.

Table 1-20 BXActivity Class XML Constants (1 of 2)

Constant	Description
Syntax	<code>public static String APPLNAME = "APPLNAME";</code>
	<code>public static String ACTIVITYNAME = "ACTIVITYNAME";</code>
	<code>public static String SUBJECT = "SUBJECT";</code>
	<code>public static String ACTION = "ACTION";</code>
	<code>public static String ORGID = "BILLERID";</code>
	<code>public static String BILLER_ID = "BILLERID";</code>
	<code>public static String ACCTID = "ACCTID";</code>
	<code>public static String CREATEDBY = "CREATEDBY";</code>
APPLNAME	<code>public static String MESSAGE = "MESSAGE";</code>
	Application tag

Table 1-20 BXActivity Class XML Constants (2 of 2)

ACTIVITYNAME	Activity tag
SUBJECT	Subject tag
ACTION	Action tag
ORGID	Organization ID tag; interchangeable with Biller ID tag
BILLER_ID	Biller ID tag; interchangeable with Organization ID tag
ACCTID	Account ID tag
CREATEDBY	Created By tag
MESSAGE	Message tag

Activity Constants

You may specify your own value to describe an action or you can use the following constants to specify a value. Table 1-21 lists the activity constants.

Table 1-21 BXActivity Class Activity Constants

Constant	Description
Syntax	<code>public status String PROCESS = "ACTIVITY_PROCESS_";</code>
	<code>public status String ACCOUNT = "ACTIVITY_ACCOUNT_";</code>
	<code>public status String DOCUMENT = "ACTIVITY_DOCUMENT_";</code>
	<code>public status String PAYMENT = "ACTIVITY_PAYMENT_";</code>
	<code>public status String CUSTOMER = "ACTIVITY_CUSTOMER_";</code>
	<code>public status String BILLER = "ACTIVITY_BILLER_";</code>
PROCESS	Process-related activity
ACCOUNT	Account-related activity
DOCUMENT	Document-related activity
PAYMENT	Payment-related activity
CUSTOMER	Customer-related activity
BILLER	Biller-related activity

NOTE

These constants describe the subject in a subject-action pair, such as `ACTIVITY_PROCESS_STARTED` where `ACTIVITY_PROCESS_` is the subject.

Action Constants

You may specify your own value to describe an action or you can use the following constants to specify a value. Table 1-22 lists the action constants.

Table 1-22 BXActivity Class Action Constants *(1 of 2)*

Constants	Description
Syntax	<code>public static String STARTED = "STARTED";</code>
	<code>public static String STOPPED = "STARTED";</code>
	<code>public static String PAUSED = "PAUSED";</code>
	<code>public static String RESUMED = "RESUMED";</code>
	<code>public static String ERROR = "ERROR";</code>
	<code>public static String INSERTED = "INSERTED";</code>
	<code>public static String DELETED = "DELETED";</code>
	<code>public static String ACTIVATED = "ACTIVATED";</code>
	<code>public static String DEACTIVATED = "DEACTIVATED";</code>
	<code>public static String VOIDED = "VOIDED";</code>
	<code>public static String VIEWED = "VIEWED";</code>
	<code>public static String PAID = "PAID";</code>
	<code>public static String UPDATED = "UPDATED";</code>
STARTED	Activity started
STOPPED	Activity stopped
PAUSED	Activity paused
RESUMED	Activity resumed
ERROR	Error occurred
INSERTED	Insert operation occurred
DELETED	Delete operation occurred
ACTIVATED	Activation operation occurred
DEACTIVATED	Deactivation operation occurred

Table 1-22 BXActivity Class Action Constants (2 of 2)

Constants	Description
VOIDED	Operation voided
VIEWED	View operation occurred
PAID	Payment operation occurred
UPDATED	Update operation occurred.

NOTE	These constants describe the action in a subject-action pair, such as <code>ACTIVITY_PROCESS_STARTED</code> where <code>STARTED</code> is the action.

BXActivityCheck Class Reference

The BXActivityCheck class checks if the activity is proper. It is used by the BXActivity class (see “BXActivity Class Reference” on page 54).

Class Definition

Table 1-23 defines the BXActivityCheck class definition.

Table 1-23 BXActivityCheck Class Definition	
Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXActivityCheck {...}</code>

Methods

This section describes the BXActivityCheck class methods. Table 1-24 lists these methods.

Table 1-24 BXActivityCheck Class Methods

Method	Description
check()	Checks that the activity entry to be logged is proper.
GetDebugLevel()	Retrieves the minimum activity debugging.
track()	Used to track a key in a tracking file.

check()

Checks that the activity entry to be logged is proper.

Syntax	<pre>public static boolean check(String <i>billerName</i>, String <i>activityName</i>, Vector <i>attArray</i>);</pre>
	<pre>public static boolean check(String <i>activityName</i>, Vector <i>attArray</i>)</pre>
	<pre>public static boolean check(String <i>billerName</i>, String <i>activityName</i>)</pre>
	<pre>public static boolean check(String <i>activityName</i>)</pre>
	<pre>public static boolean check (String <i>activityName</i>, IValList <i>inVal</i>)</pre>
Parameters	<i>billerName</i> — A string that specifies the biller name.
	<i>activityName</i> — A string that specifiesthe name of the activity that is being tracked.
	<i>attArray</i> — A vector that specifies the attributes to track. Used to ensure that mandatory attributes are entered.
Discussion	<i>inVal</i> — an IValList representing the IvalList to check against.

GetDebugLevel()

Retrieves the minimum activity debugging.

Syntax	<pre>public static int GetDebugLevel (String <i>billerName</i>;</pre>
Parameters	<i>billerName</i> — A string representing the biller name.
Discussion	

track()

Used to track a key in a tracking file.

Syntax	<code>public static boolean track (String <i>billerName</i>, IValList <i>inVal</i>, String <i>key</i>)</code>
	<code>public static boolean track (IValList <i>inVal</i>, String <i>key</i>)</code>
Parameters	<i>billerName</i> — A string that specifies the biller name.
	<i>inVal</i> — The IvalList to check against.
	<i>key</i> — A string that specifies the key to track in a tracking file.
Discussion	

BXBaseDef Class Reference

The BXBaseDef class is the base class name, type, value (NTV) look ups.

Class Definition

Table 1-25 describes the BXBaseDef class definition.

Table 1-25 BXBaseDef Class Definition	
Package	<code>com.iplanet.ecomm.billxb2b.ejb.common;</code>
Syntax	<code>public class BXBaseDef {...}</code>

Methods

This section describes the BXBaseDef class methods. Table 1-26 lists these methods.

Table 1-26 BXBaseDef Class Methods	
Method	Description
<code>init()</code>	Initializes the NTV list.

init()

Syntax	<code>public static ACNTVList init(String fileName) throws Exception;</code>
Parameters	<i>filename</i> — A string that specifies the file name.
Discussion	

BXConfig Class Reference

The BXConfig class represent data stored in the BillerXpert configuration file, `billxpert.conf` and data stored for internationalization. The file is in name, type, value (NTV) format. See the *iPlanet BillerXpert Customization Guide* for more information about this file.

Class Definition

Table 1-27 describes the BXConfig class definition.

Table 1-27 BXConfig Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXConfig {...}</code>

Methods

This section describes the BXConfig class methods. Table 1-28 lists the BXConfig class methods. You only need to use the `getProperty()` and `setProperty()` methods from this class, which retrieve and modify, respectively, the value of a property in the file.

Table 1-28 BXConfig Class Methods (1 of 2)

Method	Description
<code>getCollator()</code>	Retrieves the Collator for comparison for internationalization.
<code>getConFile()</code>	Retrieves the configuration file.
<code>getCountry()</code>	Retrieves the country code.

Table 1-28 BXConfig Class Methods (2 of 2)

Method	Description
<code>getLanguage()</code>	Retrieves the given language for internationalization.
<code>getLocale()</code>	Retrieves the Locale for internationalization.
<code>getProperty()</code>	Retrieves the value of a property from the BillerXpert configuration file.
<code>init()</code>	Loads the configuration file.
<code>setCountry</code>	Modifies the country code.
<code>setLanguage</code>	Modifies the given language for internationalization.
<code>setProperty</code>	Modifies the value of a property in the BillerXpert configuration file.

getCollator()

Retrieves the collator.

Syntax	<code>public static Collator getCollator ();</code>
Parameters	None.
Discussion	Returns the Collator for locale independent comparisons.

getConfigFile()

Retrieves the configuration file.

Syntax	<code>public static String getConfigFile ();</code>
Parameters	None.
Discussion	Returns a string representing the BillerXpert configuration file name.

getCountry()

Retrieves the country code.

Syntax	<code>public static String getCountry();</code>

Parameters	None.
Discussion	The country code US and language would represent United States English. Returns a string representing the country code.

getLanguage()

Retrieves the language for internationalization.

Syntax	<code>public static String getLanguage();</code>
Parameters	None.
Discussion	The country code US and language code “en” would represent United States English. Returns a string representing the language.

getLocale()

Retrieves the locale.

Syntax	<code>public static String getLocale ();</code>
Parameters	None.
Discussion	Returns a string representing the locale for internationalization

getProperty()

Retrieves the value of a property from the BillerXpert configuration file.

Syntax	<code>public static String getProperty(String name) throws Exception;</code>
Parameters	<i>name</i> — A string that specifies the name of the property
Discussion	The <code>getProperty()</code> method loads the BillerXpert configuration file into memory if it is not already loaded and retrieves the value specified by the string from the file. The following example retrieves the database and user names from the configuration file:
Example	<pre>try{ String database_ = BXConfig.getProperty("database_section.database"); String username_ = BXConfig.getProperty("database_section.username");</pre>

setCountry()

Modifies the country for internationalization.

Syntax	<code>public static void setCountry (String <i>theCountry</i>) throws Exception;</code>
Parameters	<i>theCountry</i> — A string that specifies the country.
Discussion	The country code US and language code “en” would represent United States English.

setLanguage()

Modifies the language for internationalization.

Syntax	<code>public static void setLanguage (String <i>theLanguage</i>) throws Exception;</code>
Parameters	<i>theLanguage</i> — A string that specifies the language.
Discussion	The country code US and language code “en” would represent United States English.

setProperty()

Modifies the value of a property in the BillerXpert configuration file.

Syntax	<code>public static void setProperty (String <i>name</i>, String <i>value</i>) throws Exception;</code>
Parameters	<i>name</i> — A string that specifies the name of the property. <i>value</i> — A string that specifies the new value.
Discussion	Loads the BillerXpert configuration file into memory if it is not already loaded and sets the specified property with the specified value. It writes the value to the configuration file and closes the file.

BXConverter Class Reference

The BXConverter class converts between an NTV (name, type, value) list and an IValList.

Class Definition

Table 1-29 describes the BXConverter class definition.

Table 1-29 BXConverter Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXConverter {...}</code>

Methods

This section describes the BXConverter class methods. Table 1-30 lists these methods.

Table 1-30 BXConverter Class Methods

Method	Description
<code>convertToNTV()</code>	Converts from an IValList to an ACNTVList.
<code>convertToValList()</code>	Converts from an ACNTVList to an IValList.

convertToNTV()

Converts an IValList to an NTV list.

Syntax	<code>public static ACNTVList convertToNTV (IValList <i>parms</i>);</code>
Parameters	<i>parms</i> — The ACNTVList to convert.
Discussion	

convertToValList()

Converts an NTV list to an IValList.

Syntax	<code>public static IValList convertToValList (ACNTVList <i>parms</i>);</code>
Parameters	<i>parms</i> —The ACNTVList to convert.
Discussion	

BXDatetime Class Reference

The BXDatetime class converts between an NTV (name, type, value) list and a IValList.

Class Description

Table 1-31 describes the BXDatetime class definition.

Table 1-31 BXDatetime Class Definition

Package	com.iplanet.ecomm.billxb2b.core;
Syntax	public class BXDatetime { ... }

Methods

This section describes the BXDatetime class methods. Table 1-32 lists these methods.

Table 1-32 BXDatetime Class Methods (1 of 2)

Method	Description
dateToISODateString()	Converts the Java.sql.date to the ISO date (yyyy-MM-dd HH:mm:ss).
dateToDateString()	Converts JAVA.sql.date to SQL92 date (yyyy-MM-dd HH:mm:ss).
dateToTimestamp()	Converts java date to timestamp format (yyyymmdd:HHmmss).
timeToDateString()	Converts JAVA.sql.time to SQL92 time (HH:mm:ss).
dateToDateString()	Converts JAVA date to SQL92 date (yyyy-MM-dd HH:mm:ss).
dateStringToDate()	Converts kiva date string to java date.
ISODateStringToDate()	Converts ISO date string (yyyy-mm-dd HH:MM:SS) to java date.
ISO8601DateStringToDate()	Converts ISO date string (yyyy-mm-dd HH:MM:SS+/-hhmm) to java date.
dateStringToShortDate()	Converts class string to MMM-dd-yyyy HH:MM:SS to java.sql.Date.
dateStringToTime()	Converts date string MMM dd yyyy HH:mm:ss: to java.sql.Time.
timeToMinutes()	Converts time to minutes.

Table 1-32 BXDatetime Class Methods (2 of 2)

Method	Description
minutesToTime()	Converts an integer value in minutes to java.sql.Time.
String toOracleDate()	Converts date string (ISO date) to Oracle date.
String toOracleTime()	Converts date string (ISO date) to Oracle date.
toOracleShortDate()	Converts date string to Oracle short date.

dateToISODateString()

Converts the Java.sql.date to the ISO date.

Syntax	public static String dateToISODateString(Date date);
Parameters	date — A Date that specifies the SQL date to be converted.
Discussion	Uses yyyy-MM-dd HH:mm:ss date format.

dateToDateString()

Converts Java.sql.date to SQL92 date.

Syntax	public static String dateToISODateString(Date date);
Parameters	date — A Date that specifies the SQL date to be converted.
Discussion	Uses yyyy-MM-dd HH:mm:ss date format. If date is null it returns null.

dateToTimestamp()

Converts java date to timestamp format.

Syntax	public static String dateToTimestamp(Date date);
Parameters	date — A Date that specifies the java date to be converted.
Discussion	Uses yyyyymmdd:HHmmss date format.

timeToDateString()

Converts Java.sql.time to SQL92 time.

Syntax	<code>public static String timeToDateString(java.sql.Time <i>time</i>);</code>
Parameters	<i>time</i> — An java.sql.Time object that specifies the Java SQL time to be converted.
Discussion	Uses HH:mm:ss time format.

dateToDateString()

Converts Kiva date string to Java date.

Syntax	<code>public static String dateToDateString(Date <i>date</i>);</code>
Parameters	<i>date</i> — A string that specifies the Kiva date string to be converted to Java date.
Discussion	Uses MMM dd yyyy HH:mm:ss date string format.

dateStringToDate()

Converts the Kiva date string to java date.

Syntax	<code>public static Date dateStringToDate(String <i>dateString</i>) throws Exception;</code>
Parameters	<i>dateString</i> — A string that specifies the Kiva date /string to be converted.
Discussion	Uses yyyy-MM-dd HH:mm:ss date string format.

ISODateStringToDate()

Converts ISO date string to java date.

Syntax	<code>public static Date ISODateStringToDate(String <i>dateString</i>) throws Exception;</code>
Parameters	<i>dateString</i> — A string that specifies the ISO date string to be converted to Java date.
Discussion	Uses yyyy-mm-dd HH:MM:SS date string format.

ISO8601DateToStringToDate()

Converts ISO8601 date string to Java date.

Syntax	<code>public static Date ISO8601DateToStringToDate(String <i>dateString</i>) throws Exception;</code>
Parameters	<i>dateString</i> — A string that specifies the ISO8601 date string to be converted to Java date.
Discussion	Uses yyyy-mm-dd HH:MM:SS+/-hhmm date string format.

dateStringToShortDate()

Converts java.sql.Date date string to short date.

Syntax	<code>public static java.sql.Date dateStringToShortDate(String <i>dateString</i>) throws Exception;</code>
Parameters	<i>dateString</i> — A string that specifies the date string to be converted to short date. The date string must be in the format MMM dd yyyy HH:mm:ss.
Discussion	Use the Gregorian calendar to determine the short date. Subtracts 1900 from the current calendar year and adds the current month and day of the month.

dateStringToTime()

Converts the Java date string to time.

Syntax	<code>public static java.sql.Time dateStringToTime(String <i>dateString</i>) throws Exception;</code>
Parameters	<i>dateString</i> — A string that specifies the date string to be converted from Java date string to time.
Discussion	

timeToMinutes()

Converts time to minutes

Syntax	<code>public static int timeToMinutes(java.sql.Time <i>time</i>) throws Exception;</code>
Parameters	<i>time</i> — A java.sql.Time object that specifies time to be converted to minutes.

Discussion

minutesToTime()

Converts minutes to java.sql.Time.

Syntax	<code>public static java.sql.Time minutesToTime(int <i>value</i>) throws Exception;</code>
Parameters	<i>value</i> — An integer that specifies the value in minutes.

Discussion

toOracleDate()

Converts the date string (ISO date) to Oracle date.

Syntax	<code>public static String toOracleDate(String <i>value</i>);</code>
Parameters	<i>value</i> — A string that specifies the value in format YYYY-mm-DD HH24:MI:SS.

Discussion

toOracleTime()

Converts the time string (ISO time) to the Oracle time.

Syntax	<code>public static String toOracleTime(String <i>value</i>);</code>
Parameters	<i>value</i> — A string that specifies the value in format HH24:MI:SS.
Discussion	Returns “TO-DATE (value, “yyyy-MM-DD HH24:MI:SS”

toOracleTime2()

Syntax	<code>public static String toOracleTime2(String <i>value</i>);</code>
Parameters	<i>value</i> — A string that specifies the value in format HH24 MI SS.

Discussion	Parses out the ‘:’ values. Used for parameters of the format :param. Returns “TO-DATE (value, ‘HH24: MI:SS’)”.

toOracleShortDate()

Syntax	<code>public static String toOracleShortDate(String value);</code>
Parameters	<i>value</i> — A string that specifies the date value in format YYYY-MM-DD.
Discussion	Returns “TO-DATE (‘value’, ‘yyyy-MM-DD’)”.

BXDBContext Class Reference

The BXDBContext class is the base class for component business object contexts. Whenever possible, use the interlace class IBXDBContext (see “IBXDBContext Class Reference” on page 72) instead of BXDBContext.

IBXDBContext Class Reference

The IBXDBContext is an interface class that supports communications between AppLogic and EJB components.

The IBXDBContext class represent a database context. The context is used to retain state information for database access and transaction management. You cannot create instances of the IBXDBContext class directly; rather, you create instances of IBXDBContext subclasses, such as BXMembershipContext for membership classes or BXBillingContext for billing classes, from your AppLogic.

Class Definition

Table 1-33 describes the IBXDBContext class definition.

Table 1-33 IBXDBContext Class Definition	
Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public interface IBXDBContext {...}</code>

Methods

This section describes the IBXDBContext methods. Table 1-34 lists these methods.

Table 1-34 IBXDBContext Class Methods

Contents	Descriptions
<code>release()</code>	Closes the database connection and releases memory for this context object.
<code>getBillerID()</code>	Determines Biller ID
<code>getCompanyID()</code>	Determines Company ID
<code>setCompanyID()</code>	Specifies Company ID
<code>getStatus()</code>	Determines the status object currently associated with this context.
<code>setStatus()</code>	Specifies the status object to associate with this context.

`release()`

Closes the database connection and releases memory for this context object.

Syntax	<code>public void release() throws Exception;</code>
Parameters	None.
Discussion	Call this method when you no longer need the database connection. Because a database connection is a shared resource, which is cached in a connection pool managed by iAS, the sooner you release your connection the sooner it can be reused.

`getBillerID()`

Determines the Biller ID.

Syntax	<code>public String BillerID();</code>
Parameters	None.
Discussion	Returns the Biller ID associated with the current context.

getCompanyID()

Determines the Company ID.

Syntax	<code>public String getCompanyID();</code>
Parameters	None.
Discussion	Returns the Company ID associated with the current context

getStatus()

Determines the status of an object associated with this context.

Syntax	<code>BXStatus getStatus();</code>
Parameters	None.
Discussion	Returns a BXStatus object that specifies the current context.

setCompanyID()

Specifies the Company ID.

Syntax	<code>public void setCompanyID (String <i>companyID</i>);</code>
Parameters	<i>CompanyID</i> — A string that specifies the company ID.
Discussion	Returns a reference to the status object.

setStatus()

Specifies the status object to associate with this context.

Syntax	<code>public BXStatus setStatus(BXStatus <i>status</i>);</code>
Parameters	<i>status</i> — A BXStatus object that specifies the status.
Discussion	Returns a reference to the status object.

BXDBDef Class Reference

The BXDBDef class holds the class definition for all the enumerated types in the base class.

BXDef Class Reference

The BXDef class contains the class definitions for all enumerated types and holds all the constants.

BXEntityObject Class Reference

The BXEntityObject class is the base class for the entity bean.

Class Definition

Table 1-35 describes the BXEntityObject class definition.

Table 1-35 BXEntityObject Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.common;</code>
Syntax	<code>public abstract class BXEntityObject implements IBXEntity {...}</code>

Fields

The BXEntityObject class contains the following fields, described in Table 1-36.

Table 1-36 BXEntityObject Class Fields

Field	Syntax	Description
<code>objID_</code>	<code>protected String objID;</code>	object ID/primary key
<code>status_</code>	<code>protected BXStatus status;</code>	Error stack
<code>stopWatch_</code>	<code>protected BXStopwatch stopWatch_</code>	new <code>BXStopwatch()</code> that is used for measurement.
<code>objMapKey_</code>	<code>protected String objMapKey;</code>	Stored parent key for object mapping

Methods

This section describes the BXEntityObject methods. Table 1-37 lists these methods.

Table 1-37 BXEntityObject Class Methods

Method	Description
<code>getPrimaryKey()</code>	Specified by <code>getPrimaryKey</code> in the interface class <code>IBXEntity</code> .
<code>setPrimaryKey()</code>	Specifies the primary key.
<code>getTime()</code>	Retrieves the time.
<code>resetTime()</code>	Resets the time.
<code>dump()</code>	Does a pretty print of the NTV list.
<code>getStatus()</code>	Retrieves the status.
<code>setStatus()</code>	Specifies the status.
<code>getObjMapKey()</code>	Retrieves the object map key.
<code>setObjMapKey()</code>	Specifies the object map key.
<code>checkPrimaryKey()</code>	Checks the primary key.
<code>getBXStatus()</code>	Retrieves the <code>BXStatus</code> .

`getPrimaryKey()`

Specified by `getPrimaryKey` in the interface class `IBXEntity`.

Syntax	<code>public String getPrimaryKey();</code>
Parameters	None.
Discussion	Returns the string <code>valueOf(ObjID)</code> .

`setPrimaryKey()`

Specifies the primary key.

Syntax	<code>public void setPrimaryKey (String <i>objID</i>);</code>
Parameters	<i>objID</i> — A string that specifies the primary key object ID.
Discussion	Returns the string <code>ObjID</code> .

getTime()

Retrieves the time.

Syntax	<code>public long getTime();</code>
Parameters	None.
Discussion	Returns stopWatch.getTime().

resetTime()

Resets the time.

Syntax	<code>public long getTime();</code>
Parameters	None.
Discussion	Returns stopWatch.getTime().

dump()

Does a pretty print of the NTV list.

Syntax	<code>public void dump (PrintStream out);</code>
Parameters	<i>out</i> — A PrintStream that specifies the NTV list output.
Discussion	Returns stopWatch.getTime().

getStatus()

Retrieves the status.

Syntax	<code>public BXStatus getStatus();</code>
Parameters	None.
Discussion	Returns status.__

setStatus()

Specifies the status.

Syntax	<code>public void setStatus(BXStatus <i>status</i>);</code>
Parameters	<i>status</i> — BXStatus status object.
Discussion	<code>status_ = status.</code>

getObjMapKey()

Retrieves the object map key.

Syntax	<code>public String getObjMapKey();</code>
Parameters	None.
Discussion	Returns <code>objMapKey._</code>

setObjMapKey()

Specifies the object map key.

Syntax	<code>public void setObjMapKey(String <i>value</i>);</code>
Parameters	<i>value</i> — A string that specifies the stored parent key for object mapping.
Discussion	<code>objMapKey_ = value</code>

checkPrimaryKey()

Checks the primary key.

Syntax	<code>protected void checkPrimaryKey() throws Exception;</code>
Parameters	None.
Discussion	

getBXStatus()

Retrieves the BXStatus.

Syntax	<code>protected BXStatus getBXStatus();</code>
Parameters	None.
Discussion	Returns status.

BXEnumDef Class Reference

The BXEnumDef class retrieves enumerator types by reading from the `enumdef.conf` configuration file.

Class Definition

Table 1-38 describes the BXEnumDef class definition.

Table 1-38 BXEnumDef Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.common;</code>
Syntax	<code>import com.iplanet.ecomm.billxb2b.core.*;</code>

Methods

This section describes the BXEnumDef class methods. Table 1-39 lists these methods.

Table 1-39 BXEnumDef Class Methods

Method	Definition
<code>init()</code>	Reads the filename <code>enumdef.conf</code> from the <code>billxpertb2b.conf</code> file.
<code>getEnum()</code>	Retrieves the enumeration type.
<code>getEnumByName()</code>	Retrieves the enumeration type by name.
<code>getEnums()</code>	Retrieves all the enumeration types.

getEnum()

Given Enum, returns an Enum (in the database format). If enum is null, returns the default

Syntax	<code>public static String getEnum (String <i>enumType</i>, String <i>enum</i>);</code>
Parameters	<i>enumType</i> — A string that specifies the enumeration type. <i>enum</i> — A string that specifies an Enum.
Discussion	

getEnumByName()

Given Enum, returns an Enum (in the database format). If enum is null, return default

Syntax	<code>public static String getEnumByName (String <i>enumType</i>, String <i>enumName</i>);</code>
Parameters	<i>enumType</i> — A string that specifies the enumeration type. <i>enumName</i> — A string that specifies the object retrieved by the enumeration name.
Discussion	

getEnums()

Retrieves all the values by name.

Syntax	<code>public static Vector getEnums (String <i>enumType</i>) throws Exception;</code>
Parameters	<i>enumType</i> — A string that specifies the enumeration type.
Discussion	

init()

Retrieves the file name from enumdef.conf from billxpertb2b.conf file.

Syntax	<code>public static void init() throws Exception;</code>
Parameters	None.

Discussion

BXError Class Reference

The BXError class represents an error. BXError implements the IBXErrorEntity interface. Stores an entity bean and loading errors as constants.

Class Definition

Table 1-40 describes the BXError class definition.

Table 1-40 BXError Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXError {...}</code>

Fields

Table 1-41 describes the BXError Class fields.

Table 1-41 BXError Class Fields

Field	Description
<code>BAD_ATTRIBUTE</code>	
<code>BAD_DB_EXCEPTION</code>	
<code>BAD_ENUMERATOR</code>	
<code>BAD_FILTER</code>	
<code>BAD_OBJ_MAP</code>	
<code>BAD_SORT</code>	
<code>BILLERXPERT_EJB_ERROR</code>	
<code>BILLERXPERT_LOADING_ERROR</code>	
<code>BXBULKLOAD_ADDOBJ_ERROR</code>	
<code>BXBULKLOAD_INSERT_ERROR</code>	
<code>BXBULKLOAD_REQ_COLUMN</code>	

Table 1-41 BXError Class Fields

Field	Description
CANNOT_FIND_PAYMENT_OBJECT	
CANNOT_GET_DB_CONNECTION	
CANNOT_GET_LDAP_CONNECTION	
CANNOT_LOAD_JDBC_DRIVER	
CANNOT_OPEN_CONFIG_FILE	

BXException Class Reference

The BXException class is the base exception class. All other exceptions should extend from this class.

Class Definition

Table 1-42 describes the BXException class definition.

Table 1-42 BXException Class Definition

Package	com.iplanet.ecomm.billxb2b.core;
Syntax	public class BXException extends Exception {...}

BXFileId Class Reference

The BXFileId class retrieves the next FileId in a sequence.

Class Definition

Table 1-43 describes the BXFileId class definition.

Table 1-43 BXFileId Class Definition

Package	com.iplanet.ecomm.billxb2b.core;
Syntax	public class BXFileId {...}

Methods

This section describes the BXFileId class methods. Table 1-44 lists these methods.

Table 1-44 BXFileId Class Methods

Methods	Description
<code>init()</code>	Creates a new billing context.
<code>getCurFileId()</code>	Returns the current place in the sequence found by the call to <code>getNewFileId()</code> .
<code>getNewFileId()</code>	Retrieves the next in the sequence for the FILEID table.

`init()`

Creates a new billing context.

Syntax	<pre>public static void init (IContext context) throws Exception; public static void init (IContext context, String billerID) throws Exception</pre>
Parameters	<i>context</i> — An IContext that specifies the context to create the billing context. <i>billerID</i> — An IContext that specifies the biller id to create the billing context.
Discussion	Passing NULL as a BillerID

`getNewFileId()`

Returns the current place in the sequence found by the call to `getNewFileId()`.

Syntax	<pre>public static int getNewFileId() throws Exception;</pre>
Parameters	None.
Discussion	Throws a new Exception if context_ is not initialized.

`getCurFileId()`

Retrieves the next in the sequence for the FILEID table.

Syntax	<pre>public static int getCurFileId();</pre>

Parameters	None.
Discussion	Returns <code>curFileId</code> .

BXFilter Class Reference

The BXFilter class is used to represent a search filter, such as the Where clause of a SQL statement or a filter list for LDAP.

The BXFilter class supports the following items in a Where clause:

- AND
- Operators: >, >=, <, <=, !=, LIKE

Class Definition

Table 1-45 describes the BXFilter class definition.

Table 1-45 BXFilter Class Definition

Package	<code>com.ipplanet.ecomm.billxb2b.ejb.common;</code>
Syntax	<code>public class BXFilter {...}</code>

Methods

This section describes the methods of the BXFilter class. Table 1-46 lists the BXFilter class methods.

Table 1-46 BXFilter Class Methods (1 of 2)

Methods	Description
<code>addCondition()</code>	Adds a condition to a filter.
<code>generateWhereClause()</code>	Generates a where clause.
<code>getBXDBContext()</code>	Returns the billing context.
<code>getConditionSegment()</code>	Retrieves the condition segment.
<code>getParams()</code>	Returns the parameters generated by the <code>getWhereClause()</code> .
<code>getWhereClause()</code>	Sets up a where clause. Replaced by <code>BXJDBCHelper</code> class.
<code>resetWhereClause()</code>	Resets the Where clause.
<code>setBXDBContext</code>	Sets the billing context.
<code>toLdapFilter()</code>	Creates an LDAP filter list.

Table 1-46 BXFilter Class Methods (2 of 2)

Methods	Description
toWhere()	Creates an SQL Where clause.

Relational Operator Constants

The following constants are used to specify a relation between an attribute of a search and a value or expression:

Constants	Description
Syntax	<code>public static final String OP_GREATER = ">";</code>
	<code>public static final String OP_GREATEREQUAL = ">=";</code>
	<code>public static final String OP_LESS = "<";</code>
	<code>public static final String OP_LESSEQUAL = "<=";</code>
	<code>public static final String OP_EQUAL = "=";</code>
	<code>public static final String OP_NOTEQUAL = "!=";</code>
	<code>public static final String OP_LIKE = "LIKE";</code>
	<code>public static final String OP_IN = "IN";</code>
	<code>public static final String COND_AND = "AND";</code>
	<code>public static final String COND_OR = "OR";</code>
OP_GREATER	Greater than
OP_GREATEREQUAL	Greater than or equal to
OP_LESS	Less than
OP_LESSEQUAL	Less than or equal to
OP_EQUAL	Equal to
OP_NOTEQUAL	Not equal to
OP_LIKE	Like
COND_AND	Conditional AND
COND_OR	Conditional OR
OP_IN	Conditional IN.

addCondition()

Adds a condition to a filter.

Syntax	<pre>void addCondition(String attrName, String op, String value);</pre> <pre>public void addCondition(String attrName, String op, String value, String condition);</pre> <pre>public void addCondition(BXFilter value, String condition);</pre>
Parameters	<i>attrName</i> — A string that specifies an attribute name. <i>op</i> — A string that specifies a relational operation. <i>value</i> — A string that specifies a value or an expression. <i>condition</i> — A string that specifies a condition.
Discussion	The <code>addCondition()</code> method adds a condition to this filter. The condition evaluates to true or false and consists of the specified attribute name compared to the specified value or expression using the specified operator. The attribute is case-insensitive. If the value parameter contains an expression, the expression is not evaluated by the <code>addCondition()</code> method. The following example sets up a filter to select only records matching the object's primary key:
Examples	<pre>try { I BXDocument doc = BXBillingMgr.createDocument(""); doc.setStatus(status); doc.setBillingContext(billingContext); BXFilter filter = new BXFilter(); Date today = new Date(); Date from = new Date(today.getYear(), today.getMonth() - numMonth, today.getDate()); String fromDate = BXDatetime.dateToDateString(from); String toDate = BXDatetime.dateToDateString(today); filter.addCondition("billdate", BXFilter.OP_GREATEREQUAL, fromDate); filter.addCondition("billdate", BXFilter.OP_LESSEQUAL, toDate); filter.addCondition("accountid", BXFilter.OP_EQUAL, acctid); docs = doc.ejbFindAll(filter, null); }</pre>

generateWhereClause()

Generates a where clause.

Syntax	<code>public void generateWhereClause (String <i>className</i>) throws Exception;</code>
Parameters	<i>className</i> — A string that specifies the class name used to retrieve the column and type in LDAP based on the class name.
Discussion	

getBXDBContext()

Returns the billing context.

Syntax	<code>public BXDBContext getBXDBContext ();</code>
Parameters	None.
Discussion	Retrieves the billing context.

getConditionSegment()

Retrieves the condition segment.

Syntax	<code>public StringBuffer getConditionSegment (String <i>className</i>) throws Exception;</code>
Parameters	<i>className</i> — A string that specifies the class name used to in a get condition segment.
Discussion	Creates an SQL condition statement such as OP_IN or OP_LIKE SQL statements.

getParams()

Retrieves the parameters set by the generateWhereClause() method.

Syntax	<code>public IList getParams ();</code>
Parameters	None.
Discussion	Returns params

getWhereClause()

Sets up the Where clause. Replaced by the BXJDBCHelper class.

Syntax	<code>public String getWhereClause();</code>
Parameters	None.
Discussion	Returns <code>where.toString()</code> . The Where clause is generated by the <code>generateWhereClause()</code> method.

generateWhereClause()

Generates a where clause.

Syntax	<code>public void generateWhereClause (String <i>className</i>) throws Exception;</code>
Parameters	<i>className</i> — A string that specifies the class name used to retrieve the column and type in LDAP based on the class name.
Discussion	Generates the where clause.

resetWhereClause()

Sets up a where clause.

Syntax	<code>public void resetWhereClause();</code>
Parameters	None.
Discussion	Resets the where clause.

setBXDBContext()

Sets up the billing context.

Syntax	<code>public void setBXDBContext (BXDBContext <i>bx</i>);</code>
Parameters	<i>bx</i> — A BXDBContext hat represents the billing context.
Discussion	

toLDAPFilter()

Creates an LDAP filter list.

Syntax	<code>ACLdapFilterList toLDAPFilter(String className) throws Exception;</code>
Parameters	<i>className</i> — A string that specifies the class to use in the filter
Discussion	The value of the <i>className</i> parameter is used to retrieve an attribute. An exception is thrown if a filter attribute is not found for the specified class. Returns an ACLdapFilterList object. The following example retrieves all descendants of the specified class. <pre>if (filter != null) { LDAPfilter = filter.toLDAPFilter(className_); }</pre>
Example	<pre>... // get all descendants. cltn = parent.getMyDescendants(dObj, LDAPfilter, LDAPorder, LDAPasc, maxRows)</pre>

toWhere()

Creates an SQL Where clause.

Syntax	<code>String toWhere(String className) throws Exception;</code>
Parameters	<i>className</i> - A string that specifies the class to use in the filter
Discussion	The Where clause created by this method does not contain the Where keyword. The value of the <i>className</i> parameter is used to retrieve an attribute. An exception is thrown if a filter attribute is not found for the specified class. If more than one condition is specified in this filter, the <code>toWhere()</code> method concatenates each condition into the Where clause using logical AND operators. Returns a string that contains an SQL Where clause.

Example

```

public void ejbRemoveAll(BXFilter filter) throws Exception {
    String where = null;
    if(filter != null) {
        try{
            filter.setBXDBContext(context_);
            //filter.generateWhereClause(getName());
            where = filter.toWhere(getName());
        } catch (Exception e) {
            getBXStatus().push(BXStatus.STATUS_ERROR,BXError.BAD_FILTER,
            null);
            throw e;
        }
    }
    stopWatch_.reset();
    stopWatch_.start();
    //Check if DBContext is valid
    checkDBContext ();
    context_.log (BXDBContext.INFO,
    "BXEventHandler::ejbRemoveAll");
    Connection conn = context_.getConnection ();
    BXJDBCHelper helper = new BXJDBCHelper (conn);
    helper.setBXDBContext (context_);
    helper.setTable (getName () );
    context_.log (BXDBContext.INFO,
    "BXEventHandler::ejbRemoveAll:after getConnection");
    //Execute the query
    context_.log (BXDBContext.INFO,
    "BXEventHandler::ejbRemoveAll:before deleteRow");
    StringBuffer strbuf = new StringBuffer ("DELETE FROM " +
    BXDBDef.BXEVENTHANDLER);
    if (where != null )

```

```
public strbuf.append ("WHERE " + where);
Hashtable rh = null;
int rc = 0;
try {
    helper.setTabledefTableName (getName () );
    rh = helper.doExecute (strbuf.toString (),
        BXJDBCHelper.DELETE);
    rc = ((Integer) rh.get(new Integer
        (BXJDBCHelper.DELETE))).intValue();
} catch (SQLException se) {
}
}
```

BXGlobal Class Reference

The BXGlobal class keeps an NTV (name, type, value) list for each biller’s table definition.

Class Definition

Table 1-47 describes the BXGlobal class definition.

Table 1-47 BXGlobal Class Definition

Package	package com.iplanet.ecomm.billxb2b.core;
Syntax	public class BXGlobal {...}

Methods

This section describes the BXGlobal class methods. Table 1-48 lists these methods.

Table 1-48 BXGlobal Class Methods (1 of 2)

Method	Description
getBillerNTVlist ()	Reads the NTV list from the billxb2b-conf.biller.TABLE_DEF section in LDAP.

Table 1-48 BXGlobal Class Methods (2 of 2)

Method	Description
<code>getDefaultConf()</code>	Returns the NTV list from the default billxb2b-conf.APPLICATION section in LDAP.
<code>setBillerNTVList()</code>	Writes to the NTV list in the billxb2b-conf.biller.TABLE_DEF section in LDAP.
<code>setBillTypeNTVList()</code>	Sets the NTV list for LDAP bxb2b-conf.biller
<code>dump()</code>	Does a pretty print of the NTV list.

getBillerNTVlist()

Reads the NTV list from the billxb2b-conf.biller.TABLE_DEF section in LDAP.

Syntax	<code>public static ACNTVList getBillerNTVList (String <i>biller</i>);</code>
Parameters	<i>biller</i> — A string that specifies the biller name.
Discussion	

getDefaultConf()

Returns the NTV list from the default billxb2b-conf.APPLICATION section in LDAP. Note: If a lookup is not found under a particular biller a LDAP default lookup is then performed under billxb2b.conf.APPLICATION.

Syntax	<code>public static ACNTVList getDefaultConf();</code>
Parameters	None.
Discussion	

setBillerNTVlist()

Writes to the NTV list in the billxb2b-conf.biller.TABLE_DEF section in LDAP.

Syntax	<code>public static ACNTVList getBillerNTVList (String <i>biller</i>);</code>
Parameters	<i>biller</i> — The biller name.

Discussion

setBillTypeNTVlist()

Sets the NTV list for LDAP bxb2b-conf.biller.

Syntax	<code>public static boolean setBillTypeNTVlist (String <i>biller</i>, ACNTVList <i>billtypeNTV</i>) ;</code>
Parameters	<i>biller</i> — The biller name. <i>billtypeNTV</i> — The biller type NTV list.
Discussion	

Fields

The BXGlobal class contains the fields listed in Table 1-49.

Table 1-49 BXGlobal Class Fields

Field	Description
Syntax	<code>public static ACNTVList objDic = null;</code> <code>public static ACNTVList configtbl = null;</code> <code>public static ACNTVList enumtbl = null;</code> <code>public static Vector sessArray = null;</code> <code>public static ACNTVList defaultConf = null;</code> <code>private static Hashtable ntvlist = null;</code>
<code>objDic</code>	Object dictionary
<code>configtbl</code>	tabledef.conf configuration file
<code>enumtbl</code>	enumdef.conf configuration file
<code>sessArray</code>	Session pool
<code>defaultConf</code>	billx-conf.APPLICATION
<code>ntvlist</code>	Keeps the NTV list for each biller's table definition.

BXJDBCHelper Class Reference

The BXJDBCHelper class is a helper class that performs JDBC inserts, deletes, and updates at the EJB level.

Class Definition

Table 1-50 describes the BXJDBCHelper class definition.

Table 1-50 BXJDBCHelper Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.common;</code>
Syntax	<code>public class BXJDBCHelper {...}</code>

Methods

This section describes the BXJDBCHelper class methods. Table 1-51 lists these methods.

Table 1-51 BXJDBCHelper Class Methods

Method	Description
<code>checkHashTab()</code>	
<code>checkStmtObj()</code>	
<code>checkString()</code>	

setBXDBContext()

Sets the billing context.

Syntax	<code>public void setBXDBContext (BXDBContext <i>bx</i>);</code>
Parameters	<i>bx</i> — The billing context.
Discussion	

getBXDBContext()

Retrieves the billing context.

Syntax `public BXDBContext getBXDBContext();`

Parameters None.

Discussion

checkStmtObj()

Helps to check the statement object.

Syntax `protected void checkStmtObj() throws SQLException;`

Parameters None.

Discussion

closeStatement()

Closes the SQL statement.

Syntax `public void closeStatement();`

Parameters None.

Discussion

setTable()

Helps to set the Table name to prepare the SQL statement.

Syntax `public void setTable (String table_Name) throws Exception;`

Parameters *table_Name* — A string that specifies the table name.

Discussion

setTable()

Helps to set the table definition name, when using filter and order.

Syntax `public void setTabledefTableName (String tdef_table_Name) throws Exception;`

Parameters *tdef_table_Name* — A string that specifies the table definition name.

Discussion

getColName()

Retrieves the volume name based on the attribute passed.

Syntax `protected String getColName (String attribute) throws Exception;`

Parameters *attribute* — A string that specifies the attribute.

Discussion

checkString()

Helps to check if the string is null or equals the empty string.

Syntax `protected boolean checkString (String str);`

Parameters *str* — A string that specifies the string to parse.

Discussion Returns false if the *str* is null or equals “”. Returns true otherwise.

SQLEscapeStr()

Helps to prepare an SQL statement by using special characters in the statement.

Syntax `protected String SQLEscapeStr (String str);`

Parameters *str* — A string that specifies the string to parse.

Discussion

checkStmtType()

Checks the statement type.

Syntax	<code>protected boolean checkStmtType (int <i>type</i>);</code>
Parameters	<i>type</i> — An integer that specifies the statement type.
Discussion	Checks if the statement type is between 1 and 4.

setString()

Helps to set the integer value to prepare the SQL statement.

Syntax	<code>public void setInt (String <i>attribute</i>, int <i>val</i>) throws Exception;</code>
Parameters	<i>attribute</i> — A string attribute. <i>val</i> — An integer that specifies the value.
Discussion	Sets the attribute as a String value <i>val</i> .

setInt()

Helps to set the string value to prepare the SQL statement.

Syntax	<code>public void setString (String <i>attribute</i>, String <i>val</i>) throws Exception;</code>
Parameters	<i>attribute</i> — A string that specifies attribute. <i>val</i> — An integer that specifies value.
Discussion	Checks that the attribute has a corresponding column in LDAP and stores the value in <i>val</i> .

setDouble()

Helps to set the double value to prepare the SQL statement.

Syntax	<code>public void setDouble (String <i>attribute</i>, double <i>val</i>) throws Exception;</code>
Parameters	<i>attribute</i> — A string that specifies the attribute. <i>val</i> — A double-precision floating number that specifies the value.

Discussion	Sets the attribute as a double value <i>val</i> .

setDate()

Helps to set the date value to prepare the SQL statement.

Syntax	<code>public void setDate (String attribute, Date val) throws Exception;</code>

Parameters	<i>attribute</i> — A string that specifies the attribute.

	<i>val</i> — An integer that represent the value.

Discussion	The date will be sent in the format “YYYY-MM-dd HH:mm:ss”.

checkHashTab()

Checks if the hash table is empty.

Syntax	<code>protected Hashtable checkHashTab (Hashtable ht) throws Exception;</code>

Parameters	<i>ht</i> — A Hashtable object that specifies the hash table.

Discussion	Throws an exception if the hash table is empty.

getInsertQuery()

Prepares to insert the SQL query statement.

Syntax	<code>protected String getInsertQuery() throws Exception;</code>

Parameters	None.

Discussion	

insert()

Helps perform the insert transactions with the database.

Syntax	<code>public int insert() throws SQLException, Exception;</code>

Parameters	None.

Discussion

delQuery()

Helps prepare the Delete SQL statement query.

Syntax `protected String getDelQuery() throws Exception;`

Parameters None.

Discussion

delete()

Helps perform the delete transactions with the database.

Syntax `public int delete() throws SQLException, Exception;`

Parameters None.

Discussion Returns an integer that is the number of deleted columns.

getUpdateQuery()

Helps prepare the update SQL statement.

Syntax `protected String getUpdateQuery(String pk_val) throws Exception;`

Parameters *pk_val* — A string that specifies the primary key value used for updatring SQL query statements.

Discussion

update()

Perform an update SQL transaction on the base of the database primary key.

Syntax `public int update (String pk_val) throws SQLException, Exception;`

Parameters *pk_val* — A string that specifies the primary key value used for updatring SQL query statements.

Discussion	Returns an integer that is the number of updated columns.

getReadQuery()

Helps to prepare the select SQL query statement.

Syntax	protected String getReadQuery BXFilter <i>filter</i> , BXOrder <i>order</i> , String <i>start_index</i>) throws Exception; <i>filter</i> — A BXFilter filter for the the read query. <i>order</i> — A BXOrder order for the the read query.
Parameters	<i>start_index</i> — A string that specifies the starting index for sequence numbers. NOTE: the BillerXpert Oracle schema frequently starts with an index of 100 or great. Customers should start their Oracle indexes at 1000 or greater both for performance tuning and if index entries need to be manually entered some time in the future.
Discussion	The BXFilter and BXOrder parameters help in Where conditions.

getReadCounterQuery()

Helps prepare the select SQL query statement and sets the index counter.

Syntax	protected String getReadCounterQuery (BXFilter <i>filter</i> , BXOrder <i>order</i> , String <i>start_index</i>) throws Exception; <i>filter</i> — A BXFilter filter for the the read query. <i>order</i> — A BXOrder order for the the read query.
Parameters	<i>start_index</i> — A string that specifies the starting index for sequence numbers. NOTE: the BillerXpert Oracle schema frequently starts with an index of 100 or great. Customers should start their Oracle indexes at 1000 or greater both for performance tuning and if index entries need to be manually entered some time in the future.
Discussion	

getSqlFilterOrder()

Helps to get SQL statement filter order.

Syntax	public String getSqlFilterOrder (BXFilter <i>filter</i> , BXOrder <i>order</i> , String <i>str</i>) throws Exception;

	<i>filter</i> — A BXFilter filter for the the read query.
Parameters	<i>order</i> — A BXOrder order for the the read query.
	<i>str</i> — A string that specifies SQL query in string form that you can append to.
Discussion	

read()

Helps select transactions and read the date from the database.

Syntax	<code>public ResultSet read (BXFilter <i>filter</i>, BXOrder <i>order</i>, String <i>start_index</i>, int <i>max_index</i>) throws SQLException, Exception;</code>
	<i>filter</i> — A BXFilter filter for the the read query.
Parameter	<i>order</i> — A BXOrder order for the the read query.
	<i>start_index</i> — A string that specifies the starting index for sequence numbers.
	<i>max_index</i> — An integert that specifies the maximum index.
Discussion	Returns ResultSet.

readCounter()

Helps count the transactions and read the data from the database.

Syntax	<code>public int readCounter(BXFilter <i>filter</i>, BXOrder <i>order</i>, String <i>start_index</i>, int <i>max_index</i>) throws SQLException, Exception;</code>
	<i>filter</i> — A BXFilter filter for the the read query.
Parameters	<i>order</i> — A BXOrder order for the the read query.
	<i>start_index</i> — A string that specifies the starting index for sequence numbers.
	<i>max_index</i> — An integert that specifies the maximum index.
Discussion	

read()

Helps select transactions and read the data from the database.

Syntax	<code>public ResultSet read() throws SQLException, Exception;</code>

Parameters None.

Discussion

getSqlQuery()

Retrieves the SQL query.

Syntax	<pre>public String getSqlQuery(BXFilter <i>filter</i>, BXOrder <i>order</i>, String <i>pk_val</i>, String <i>start_index</i>, int <i>type</i>) throws SQLException, Exception;</pre> <i>filter</i> — A BXFilter filter for the the read query. <i>order</i> — A BXOrder order for the the read query.
Parameters	<i>start_index</i> — A string that specifies the starting index for sequence numbers. <i>max_index</i> — An integert that specifies the maximum index. <i>type</i> —An integer that resrepresents the SQL statement type
Discussion	Returns the string SQL query.

doExecute()

Executes an SQL query and retrieves an appropriate type (result set or update count) from the database and puts it into the hash table with the key value (either statement type, result set, or integer).

Syntax	<pre>public Hashtable doExecute (BXFilter <i>filter</i>, BXOrder <i>order</i>, String <i>query</i>, int <i>maxRows</i>, int <i>type</i>) throws SQLException, Exception;</pre> <i>filter</i> — A BXFilter filter for the the read query. <i>order</i> — A BXOrder order for the the read query.
Parameters	<i>query</i> — A string that specifies the query statement. <i>maxRows</i> — An integer that specifies the maximum rows in the query statement. <i>type</i> — An integer that resrepresents the SQL statement type.
Discussion	While taking the values from the hash table, this method checks the statement type. If the statement type is ==1 the integer value type is result set. If type >= 4 the value is the update count.

doExecuteGroupBy()

Executes a group SQL query and retrieves an appropriate type (result set or update count) from the database and puts it into the hash table with the key value (either statement type, result set, or integer).

Syntax	<pre>public Hashtable doExecuteGroupBy (BXFilter filter, BXOrder order, String query, int maxRows, int type) throws SQLException, Exception;</pre> <i>filter</i> — A BXFilter filter for the the read query. <i>order</i> — A BXOrder order for the the read query.
Parameters	<i>query</i> — A string that specifies the query statement. <i>maxRows</i> — An integer that specifies the maximum rows in the query statement. <i>type</i> — An integer that represents the SQL statement type.
Discussion	While taking the values from the hash table, this method checks the statement type. If the statement type is ==1 the integer value type is result set. If type >= 4 the value type is the update count.

doExecute()

Executes an SQL query and retrieves the statement type.

Syntax	<pre>public Hashtable doExecute (String query, int type) throws SQLException, Exception;</pre>
Parameters	<i>query</i> — A string that specifies the query statement. <i>type</i> — An integer that represents the SQL statement type
Discussion	While taking the values from the hash table, this method checks the statement type. If the statement type is ==1 the integer value type is the result set. If type >= 4 the value is the update count.

BXMessages Class Reference

The BXMessages class reads the messages and the local information from the `billx2b.conf` configuration file.

NOTE	The locale information has been obsoleted in the BXMessages class. Please refer to the BXConfig class (see “BXConfig Class Reference” on page 62).

Class Definition

Table 1-52 describes the BXMessages class definition.

Table 1-52 BXMessages Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXMessages {...}</code>

Methods

This section describes the BXMessages class methods. Table 1-53 lists these methods.

Table 1-53 BXMessages Class Methods

Method	Description
<code>dump()</code>	Dumps messages from the message table for debugging.
<code>init()</code>	Retrieves the message file name from the <code>billx2b.conf</code> configuration file.

`dump()`

Dumps messages from the message table for debugging.

Syntax	<code>public static void dump (ACNTVList <i>ntvList</i>, String <i>prefix</i>);</code> <i>ntvList</i> — The NTV (name, type, value) list.
Parameters	<i>prefix</i> — The prefix used to pretty print the NTV lists. The standard NTV list call <code>ntvList.doprettyprint()</code> is used.
Discussion	

`init()`

Retrieves the message file name from the `billx2b.conf` configuration file.

Syntax	<code>public static void init();</code>
Parameters	None.

Discussion	This message reads the location of the message file and the locale information from the <code>billxper.conf</code> configuration file.

BXObjectMgr Class Reference

The BXObjectMgr class resolves the mapping between the interface name and the actual class implementation, depending upon the biller name and the bill type. The biller name is added to make the class work for different billers. The bill type is added to make the class work for a biller that has multiple bill type.

Class Definition

Table 1-54 describes the BXObjectMgr class definition.

Table 1-54 BXObjectMgr Class Definition	
Package	<code>com.ipplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXObjectMgr {...}</code>

Methods

This section describe the BXObjectMgr class method. Table 1-55 lists this method.

Table 1-55 BXObjectMgr Class Method	
Method	Description
<code>createObject()</code>	Returns an object that implements the given interface name, biller name, bill type, and account type.

createObject()

Returns an object that implements the given interface, billerName, billType, and actual class implementation.

Syntax	<code>public static Object createObject (String <i>interfaceName</i>, String <i>billerName</i>, String <i>billType</i>, String <i>acctType</i>) throws Exception;</code>

Parameters	<i>interfaceName</i> — A string that specifies the given interface name.
	<i>billerName</i> — A string that specifies the biller name.
	<i>billType</i> — A string that specifies the bill type.
	<i>acctType</i> — A string that specifies the actual class implementation.
Discussion	The <i>billerName</i> and <i>billType</i> parameters can be null, but the <i>interfaceName</i> parameter should not be null.

BXObjMap Class Reference

This BXObjMap class is used by the BXObjectMgr class (see BXObjectMgr Class Reference). The BXObjMap class resolves the mapping between the interface and the actual class implementation, depending upon the biller name and the bill type.

Class Definition

Table 1-56 describes the BXObjMap class definition.

Table 1-56 BXObjMap Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXObjMap {...}</code>

Methods

This section describes the BXObjMap class methods. Table 1-57 lists these methods.

Table 1-57 BXObjMap Class Methods

Method	Description
<code>ObjMap()</code>	

ObjMap()

Used by BXObjectMgr to map the interface and the actual class.

	<code>public static Class ObjMap(String <i>key</i>, String <i>object</i>, ACStatus <i>status</i>) throws Exception;</code>
Syntax	<code>public static Class ObjMap (String <i>object</i>, ACStatus <i>status</i>) throws Exception;</code> <code>public static Class ObjMap (String <i>object</i>) throws Exception;</code> <i>key</i> — A string that specifies the biller name.
Parameters	<i>object</i> — A string that specifies the interface name. <i>status</i> — An ACStatus that specifies the status.
Discussion	The class is returned.

BXOrder Class Reference

The BXOrder class represents order in which search results are returned, such as the Order By clause of an SQL statement or the order associated with attributes in an LDAP filter list.

Class Definition

Table 1-58 describes the BXOrder class definition.

Table 1-58 BXOrder Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.common;</code>
Syntax	<code>public class BXOrder {...}</code>

Order Specification Constants

The following constants are used to specify whether a record will be retrieved in ascending or descending order of the associated attribute:

Syntax	<code>public static final String ASC = "ASC";</code>
	<code>public static final String DSC = "DSC"</code>
	ASC - Ascending order
	DSC - Descending order

Methods

This section describes the BXOrder class methods. Table 1-59 lists these methods.

Table 1-59 BXOrder Class Methods (1 of 2)

Methods	Description
<code>addOrder()</code>	Associates an order with an attribute.
<code>resetOrderBy()</code>	Resets the SQL Order By clause.
<code>getSize()</code>	Determines the number of attributes for which an ordering has been specified.
<code>toLdapOrderBy()</code>	Creates an ordering for attributes in an LDAP filter list.

Table 1-59 BXOrder Class Methods (2 of 2)

Methods	Description
toOrderBy()	Creates an SQL Order By clause.
setDBContext()	Sets the database (DB) context.
getDBContext()	Retrieves the database (DB) context.

addOrder()

Associates an order with an attribute.

Syntax	<pre>public void addOrder (String attrName, String order);</pre>
Parameters	<i>attrName</i> — A string that specifies an attribute name. <i>order</i> — A string that specifies the order.
Discussion	The addOrder() method adds a condition to this filter. The condition evaluates to true or false and consists of the specified attribute name compared to the specified value or expression using the specified operator. The attribute must exist in the TypeDefs.conf configuration file. If the value parameter contains an expression, the expression is not evaluated by the addCondition() method. For example: <pre>order = new BXOrder(); order.addOrder("CreateDate", BXOrder.DSC);</pre>

getSize()

Determines the number of attributes for which an ordering has been specified.

Syntax	<pre>public int getSize();</pre>
Parameters	None.
Discussion	Returns an integer that contains the name of the entity's class.

toLDAPOrderBy()

Creates an ordering for attributes in an LDAP filter list.

Syntax	<code>void toLDAPOrderBy(String className, String[] order, boolean[] asc) throws Exception;</code>
	<i>className</i> — A string that specifies the class to which the ordering attributes belong.
Parameters	<i>order</i> — An array of strings that specifies the attributes on which to order the selected records.
	<i>asc</i> — A string of booleans, each of which is true if the corresponding attribute is to be ordered in ascending order or is false if it is to be sorted in descending order
Discussion	Populates the <i>order</i> and boolean arrays with the values specified by calling the <code>addorder()</code> method. The elements of these arrays appear in the same order they were specified by calling the <code>addorder()</code> method. The value of the <i>className</i> parameter is used to retrieve an attribute. An exception is thrown if a filter attribute is not found for the specified class.

toOrderBy()

Creates an SQL Order By clause.

Syntax	<code>String toOrderBy(String className) throws Exception;</code>
Parameters	<i>className</i> — A string that specifies the class to use in the filter.
Discussion	The attributes are returned as pairs consisting of an attribute and an ordering (ascending or descending); each pair is separated by a comma. The pairs appear in the same order as specified by calling the <code>addOrder()</code> method. The Order By clause created does not contain the Order By keywords. The value of the <i>className</i> parameter is used to retrieve an attribute. An exception is thrown if an order by attribute is not found for the specified class. Returns a string that contains an SQL order by clause.

Example

```
String orderBy = null;
if( order != null ) {
try {
orderBy = order.toOrderBy( getName() );
} catch ( Exception e ) {
getBXStatus().push( BXStatus.STATUS_ERROR,
BXError.BAD_SORT, null);
throw e;
}
}
...
IQuery query = context_.createQuery();
...
if( orderBy != null )
query.setOrderBy( orderBy );
```

resetOrderBy()

Resets the Order By clause.

Syntax	public void resetOrderBy();
Parameters	None.
Discussion	Reset the Order By clause.

setDBContext()

Sets the database context.

Syntax	public void set DBContext (BXDBContext bx);
Parameters	bx — A BXDBContext that specifies the database context.
Discussion	

getDBContext()

Retrieves the database context.

Syntax	public void get DBContext (BXDBContext bx);

Parameters	<i>bx</i> — A BXDBContext that specifies the database context.
Discussion	The database context.

BXStatus Class Reference

The BXStatus class represents a stack of status messages. You call the `push()` method to push an ID onto the object's stack. BillerXpert provides a BXStatus object you can use or you can create your own object.

If you use BillerXpert's BXStatus object, BillerXpert pops messages off the stack and processes them for you by replacing the ID with an actual message. You only need to use the following methods to retrieve information from the message stack.

Class Definition

Table 1-60 describes the BXStatus class definition.

Table 1-60 BXStatus Class Definition

Package	<code>com.ipplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXStatus {...}</code>

Constants

This section describes the constants contained in the BXStatus class. You may specify the kind of status to report by using a status constants.

Result Code Constants

Table 1-61 lists the result code constants.

Table 1-61 BXStatus Class Result Code Constants

Constant	Description
Syntax	<pre>public static final int STATUS_OK = ACStatus.ACSTATUS_OK ; public static final int STATUS_INFO = ACStatus.ACSTATUS_INFO ; public static final int STATUS_WARNING = ACStatus.ACSTATUS_WARNING ; public static final int STATUS_ERROR = ACStatus.ACSTATUS_ERROR; public static final int STATUS_FATAL_ERROR = ACStatus.ACSTATUS_FATAL_ERROR;</pre>
STATUS_OK	No error
STATUS_INFO	Informational message
STATUS_WARNING	Warning message
STATUS_ERROR	Error message
STATUS_FATAL_ERROR	Unknown fatal error

Name Constants

You may retrieve a value from the IVallList associated with the status object by specifying the corresponding name constant.

Table 1-62 BXStatus Class Name Constants (1 of 2)

Constant	Description
Syntax	<pre>public static final String RESULT_CODE = "resultCode"; public static final String MESSAGE_CODE = "messageCode"; public static final String MESSAGE_TYPE = "messageType"; public static final String MESSAGE_FORMAT = "messageFormat"; public static final String MESSAGE_TEXT = "messageText"; public static final String EXCEPTION_DATA = "exceptionData"; public static final String SOURCE_FILE = "sourceFile"; public static final String SOURCE_LINE = "sourceLine";</pre>
RESULT_CODE	Result code
MESSAGE_CODE	Message code
MESSAGE_TYPE	Message type

Table 1-62 BXStatus Class Name Constants (2 of 2)

Constant	Description
MESSAGE_FORMAT	Message format
MESSAGE_TEXT	Message text
RESULT_CODE	Result code
EXCEPTION_DATA	Exception data
SOURCE_FILE	Source file

Miscellaneous Constants

The following constants specify values in the status list that indicate whether the status information should be logged or displayed. Table 1-64 lists the miscellaneous constants.

Table 1-63 BXStatus Class Miscellaneous Constants

Constant	Description
Syntax	<pre>public static final String IS_LOG = "isLog"; public static final String IS_DISPLAY= "isDisplay";</pre>
Context	IS_LOG —A string or object that is logged. IS_DISPLAY — A string or object that is displayed

Constructor

The BXStatus class contains the following constructor.

Syntax	<pre>public BXStatus(); public BXStatus (IValList[] list);</pre>
Parameters	<i>list</i> — An array of IValList objects whose elements are error messages.
Discussion	Creates a BXStatus object. Use the first form of the constructor if you only need to specify the message when the object is pushed onto the stack. Use the second form of the constructor when you want to create an object that contains one or more messages.

Methods

This section describes the BXStatus class methods. Table 1-64 lists these methods.

Table 1-64 BXStatus Class Methods

Method	Description
<code>BXStatus()</code>	Creates a BXStatus object
<code>getBillerID()</code>	Specifies the BillerID
<code>setBillerID()</code>	Retrieves the BillerID
<code>push()</code>	Pushes a message on to the status object's message stack.
<code>pop()</code>	Pops the next message from the status object's message stack.
<code>top()</code>	Retrieve the first message from the status object's message stack.
<code>peek()</code>	Retrieves a message from the status object's message stack.
<code>process()</code>	Decodes and formats a message.
<code>count()</code>	Determines the number of messages in the status object's message stack.
<code>getACStatus()</code>	Retrieves the ACStatus object
<code>dump()</code>	Pretty prints the status information
<code>clear()</code>	Remove all messages from in the status object's message stack.

clear()

Removes all messages from in the status object's message stack.

Syntax	<code>public void clear();</code>
Parameters	None.
Discussion	

count()

Determines the number of messages in the status object's message stack.

Syntax	<code>public int count ();</code>
Parameters	None.
Discussion	Returns an integer that returns the number of messages on the stack. If there are no messages on the stack, the count () method returns 0.

dump()

Pretty prints the status information.

Syntax	<code>public void dump ();</code>
Parameters	None.
Discussion	

getACStatus()

Retrieves the ACStatus object.

Syntax	<code>public ACStatus getACStatus ();</code>
Parameters	None.
Discussion	

getBillerID()

Retrieves the Biller ID.

Syntax	<code>public String BillerID ();</code>
Parameters	None.
Discussion	

peek()

Retrieves a message from the status object's message stack.

Syntax	<code>public IList peek(int pos);</code>
Parameters	<i>pos</i> — An integer that specifies the position of message to retrieve.
Discussion	Returns an IList object that contains the next message or null. The message stack is not modified; the pointer to the next message does not change. If a message does not exist at the specified position, null is returned. Messages begin at position 0.

pop()

Pops the next message from the status object's message stack.

Syntax	<code>public void pop();</code>
Parameters	None.
Discussion	The pop() method causes the top of the stack to change. The following method processes and pops all messages for a status object:
Example	<pre>protected void logErrorMsg(BXStatus status) { if(status == null) { logOutput("Status is not set "); return; } status.process(); IList elm; while ((elm=status.top())!= null) { int rcd=elm.getValInt(BXStatus.RESULT_CODE); String msgcd=elm.getValString(BXStatus.MESSAGE_CODE); String msg=elm.getValString(BXStatus.MESSAGE_TEXT); logOutput("Result code = "+rcd+", Error number = "+msgcd+", error msg= "+msg); status.pop(); // get rid of the processed one } }</pre>

```

protected void logErrorMsg( BXStatus status ) {
    if( status == null ) {
        logOutput("Status is not set " );
        return;
    }
    status.process();
    IValList elm;
    while ( (elm=status.top() )!= null) {
        int rcd=elm.getValInt(BXStatus.RESULT_CODE);
        String
        msgcd=elm.getValString(BXStatus.MESSAGE_CODE);
        String
        msg=elm.getValString(BXStatus.MESSAGE_TEXT);
        logOutput("Result code = "+rcd+",
        Error number = "+msgcd+", error msg= "+msg );
        status.pop(); // get rid of the processed one
    }
}

```

process()

Decodes and formats a message.

Syntax	<code>public void process();</code>
Parameters	None.
Discussion	See the example for the pop() method.

push()

Pushes a message on to the status object's message stack.

Syntax	<pre> public void push (int <i>resultcd</i>, String <i>msgcd</i>); public void push (int <i>resultcd</i>, String <i>msgcd</i>, IValList <i>parms</i>); public void push (int <i>resultcd</i>, String <i>msgcd</i>, IValList <i>parms</i>, boolean <i>display</i>, boolean <i>log</i>); public void push (int <i>resultcd</i>, int <i>errCd</i>, String <i>errMsg</i>, String <i>traceMsg</i>, boolean <i>display</i>, boolean <i>log</i>); </pre>

	<i>resultcd</i> — A string that specifies the result code constant. <i>msgcd</i> — A string that specifies the message code. <i>parms</i> — An <i>IValList</i> object that specifies the message text. <i>traceMsg</i> — A string that specifies an additional trace message.
Parameters	<i>display</i> — A boolean value that specifies whether to display the message. <i>log</i> — A boolean value that specifies whether to log the message <i>errCd</i> — A string that specifies the error code. <i>errMsg</i> — A string that specifies the error message.
Discussion	Use the first form of the <code>push()</code> method to both log and display the message. Use the second form the <code>push()</code> method to specify values for token replacement. Use the third form of the method to explicitly specify whether to log the message or display the message. The following example pushes a message onto the <i>BillerXpert</i> defined status object's stack: <pre>if (request == null) { status.push(BXStatus.STATUS_ERROR, BXErrorCode. AL_INVALID_REQUEST); }</pre>
Example	The following example pushes a message with one token onto the <i>BillerXpert</i> defined status object's stack: <pre>IValList paramList = GX.CreateValList(); paramList.setValString("function", "ejbFindAll"); getBXStatus().push(BXStatus.STATUS_ERROR, BXError.NOT_IMPLEMENTED, paramList);</pre> The message associated with <code>BXError.NOT_IMPLEMENTED</code> is: <pre>"EJB_000018" Str "[@function@] is not implemented."</pre>

setBillerID()

Manipulates the *Biller ID*.

Syntax	<pre>public void setBillerID (String <i>billerID</i>);</pre>
Parameters	<i>billerID</i> — A string that specifies the <i>BillerID</i>

top()

Retrieves the first message from the status object's message stack.

Syntax	<code>public IList top();</code>
Parameters	None.
Discussion	Returns an IList object that contains the first message or null. If the stack is empty; null is returned. See the example for “pop()” on page 118.

BXStopwatch Class Reference

The BXStopwatch class represents the elapsed time, in milliseconds, as a string. This class is used for performance testing at the EJB level.

Class Definition

Table 1-65 describes the BXStopwatch class definition.

Table 1-65 BXStopwatch Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.core;</code>
Syntax	<code>public class BXStopwatch {...}</code>

Methods

This section describes the BXStopwatch class methods. Table 1-66 lists these methods.

Table 1-66 BXStopwatch Class Methods

Method	Description
<code>start()</code>	Begins timing.
<code>start()</code>	Begins timing and uses the <i>elapsed</i> parameter.
<code>stop()</code>	Stops timing.
<code>reset()</code>	Resets the timer.
<code>active()</code>	Activates the timer.

Table 1-66 BXStopwatch Class Methods

Method	Description
toString()	Returns the current elapsed time in milliseconds.
getTime()	Retrieves the current elapsed time and the current system time in milliseconds.
finalize()	Overrides the finalize() method in the java.lang.Object.

start()

Begins timing.

Syntax	public void start();
Parameters	None.
Discussion	

start()

Begins timing.

Syntax	public void start(long <i>elapsed</i>);
Parameters	<i>elapsed</i> — A long that specifies the elapsed time
Discussion	

stop()

Stops timing.

Syntax	public void stop();
Parameters	None.
Discussion	

reset()

Resets the timer.

Syntax	<code>public void reset();</code>

Parameters	None.

Discussion	

active()

Activates the timer.

Syntax	<code>public boolean active();</code>

Parameters	None.

Discussion	

toString()

Returns the current elapsed time in milliseconds as a string.

Syntax	<code>public String toString();</code>

Parameters	None.

Discussion	Overrides the <code>toString()</code> method in <code>java.lang.Object</code> .

getTime()

Retrieves the current slapped time and the current system time in milliseconds.

Syntax	<code>public long getTime();</code>

Parameters	None.

Discussion	

finalize()

Overrides the `finalize()` method in the `java.lang.Object`.

Syntax	<code>public void finalize();</code>
Parameters	None.
Discussion	

BXTableDef Class Reference

The `BXTableDef` class performs a lookup to match the Oracle table and column names to fields stored in LDAP. This means the mapping of these fields is independent of the Oracle table and column names, in case they change in the future.

Class Definition

Table 1-67 describes the `BXTableDef` class definition.

Table 1-67 BXTableDef Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.common;</code>
Syntax	<code>public class BXTableDef {...}</code>

Methods

This section describes the `BXTableDef` class methods. Table 1-68 lists these methods.

Table 1-68 BXTableDef Class Methods

Method	Description
<code>init()</code>	Retrieves the file name for <code>tabledef.conf</code> from the configuration file.
<code>getTableName()</code>	Retrieves the table name.
<code>getColumnName()</code>	Retrieves the column name in the table.
<code>getAttrType()</code>	Retrieves the attributes for the table.

Table 1-68 BXTableDef Class Methods

Method	Description
<code>getOrdinal()</code>	Retrieves the ordinal position in the table.
<code>getLength()</code>	Retrieves the column length in the table.

`init()`

Retrieves the file name from the `billerxper.conf` configuration file.

Syntax	<code>public static void init() throws Exception;</code>
Parameters	None.
Discussion	

`getTableName()`

Retrieves the table name for the specified class.

Syntax	<code>public static String getTableName (String <i>className</i>) throws Exception;</code> <code>public static String getTableName (String <i>biller</i>, String <i>className</i>) throws Exception;</code>
Parameters	<i>biller</i> — A string that specifies the biller name. <i>className</i> — A string that specifies the class name.
Discussion	

getColumnName()

Retrieves the column name in the table for the specified class.

Syntax	<pre>public static String getColumnName (String className, String attrName) throws Exception;</pre>
	<pre>public static String getColumnName (String biller, String className, String attrName) throws Exception;</pre>
	<pre>public static String getColumnName (String className, String attrName, boolean mixcase) throws Exception;</pre>
	<pre>public static String getColumnName (String biller, String className, String attrName, boolean mixcase) throws Exception;</pre>
Parameters	<i>biller</i> — A string that specifies the biller name.
	<i>className</i> — A string that specifies the class name that maps to this column name.
	<i>attrName</i> — A string that specifies the attribute name.
	<i>mixcase</i> — Not currently used because column names are not case sensitive. Note: This parameter is retained to ensure <i>biller</i> and <i>attrName</i> backward compatibility for current customers.
Discussion	

getAttrType()

Retrieves the attribute type for the specified class.

Syntax	<pre>public static String getAttrType (String className, String attrName) throws Exception;</pre>
	<pre>public static String getAttrType (String biller, String className, String attrName) throws Exception;</pre>
	<pre>public static String getAttrType (String className, String attrName, String mixcase) throws Exception;</pre>
	<pre>public static String getAttrType (String biller, String className, String attrName, boolean mixcase) throws Exception;</pre>
Parameters	<i>biller</i> — A string that specifies the biller name.
	<i>className</i> — A string that specifies the class name that maps to this column name.
	<i>attrName</i> — A string that specifies the attribute name.
	<i>mixcase</i> — Not currently used because column names are not case sensitive. Note: This parameter is retained to ensure <i>biller</i> and <i>attrName</i> backward compatibility for current customers.

Discussion

getOrdinal()

Retrieves the ordinal position in the table for the specified class and attribute name.

Syntax

```
public static int getOrdinal ( String className, String attrName ) throws  
Exception;  
  
public static int getOrdinal ( String biller, String className, String  
attrName ) throws Exception;
```

biller — A string that specifies the biller name.

Parameters

className — A string that specifies the class name that maps to this column name.
attrName — A string that specifies the attribute name.

Discussion

getLength()

Retrieves the column length for the specified class.

Syntax

```
public static int getLength ( String className, String attrName ) throws  
Exception;  
  
public static int getLength ( String biller, String className, String  
attrName ) throws Exception;
```

biller — A string that specifies the biller name.

Parameters

className — A string that specifies the class name that maps to this column name.
attrName — A string that specifies the attribute name.

Discussion

IBXEntity Class Reference

The IBXEntity class is an interface class for the Entity java bean. The IBXEntity subclasses represent records in a data repository. BillerXpert provides subclasses of the IBXEntity class for manipulation of these records.

You do not create IBXEntity objects directly; rather, you create instances from the BillerXpert-provided subclass. You may derive your own subclass from the IBXEntity class if necessary.

Class Definition

Table 1-69 describes the IBXEntity class definition.

Table 1-69 IBXEntity Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.common;</code>
Syntax	<code>public interface IBXEntity {...}</code>

Methods

This section describes the methods of the IBXEntity class. Table 1-70 lists these methods.

Table 1-70 IBXEntity Class Methods

Methods	Description
Getting and setting record keys	
<code>getPrimaryKey()</code>	Determines the primary key associated with a record.
<code>setPrimaryKey()</code>	Specifies the primary key associated with a record.
Database Access	
<code>ejbCreate()</code>	Creates a record in a data repository.
<code>ejbStore()</code>	Updates a record in a data repository.
<code>ejbRemove()</code>	Removes a record from a data repository.
<code>ejbLoad()</code>	Retrieves a record identified by key, from a data repository
<code>ejbFindByPrimaryKey()</code>	Retrieves a record, identified by key, from a data repository.
<code>ejbFindAll()</code>	Retrieves selected records from a data repository.
Determining the class name	
<code>getName()</code>	Determines the class name.
Getting and setting the status	

Table 1-70 IBXEntity Class Methods

Methods	Description
<code>getStatus()</code>	Determines the status object currently associated with this entity.
<code>setStatus()</code>	Specifies the status object to associate with this entity.
Debugging Support	
<code>getTime()</code>	Determine the elapsed time since the object was created.
<code>dump()</code>	Displays the fields in an object.

dump()

Displays the fields in an object.

Syntax	<code>void dump(PrintStream out);</code>
Parameters	<code>out</code> — A <code>PrintStream</code> object that specifies where to display the output.
Discussion	Subclasses of the <code>IBXEntity</code> must implement this method; there is no default implementation. Each subclass implementation should return the name of the subclass.
	The following implementation of the <code>dump</code> method displays the object ID first. <pre>public void dump(PrintStream out) { out.println("ObjectID = " + getPrimaryKey()); }</pre>

ejbCreate()

Creates a record in a data repository.

Syntax	<code>String ejbCreate() throws Exception;</code>
Parameters	None.
Discussion	The <code>ejbCreate()</code> method adds the record to the database and sets the primary key to a unique object ID by which the record can be retrieved, updated, or deleted. The database context must have been set by calling the appropriate method, such as <code>createBillingContext()</code> for billing entity objects. Subclasses of the <code>IBXEntity</code> class must implement this method because there is no default implementation.

ejbFindAll()

Retrieves selected records from a data repository.

Syntax	<pre>public Vector ejbFindAll(BXFilter <i>filter</i>, BXOrder <i>order</i>) throws Exception;</pre> <pre>public Vector ejbFindAll(BXFilter <i>filter</i>, BXOrder <i>order</i> , String <i>startIndex</i>, int <i>maxRows</i>) throws Exception;</pre>
Parameters	<i>filter</i> — A BXFilter object that specifies the Where clause of an SQL select statement. <i>order</i> — A BXOrder object that specifies the Order By clause of an SQL select statement. <i>startIndex</i> — A string that specifies the record object ID from which to start the retrieval. <i>maxRows</i> — An integer that specifies the maximum number of records to return.
Discussion	Retrieves all records that match the clause in the filter parameter, ordered by the clause specified in the order parameter. For each record, this method creates an object, sets the fields in the object from the record, and adds the object to the return vector. The first form of the syntax retrieves all records matching the filter. The second form of the syntax allows you to control the number of records returned. If the <i>filter</i> parameter is null, all records will be retrieved, up to the limit specified by the maxRows parameter. If the <i>order</i> parameter is null, the retrieved records will be sorted in ascending order by their primary key. If the <i>startIndex</i> parameter is null, the parameter is ignored. If the <i>startIndex</i> parameter is not null, all records greater than the specified value will be retrieved, up to the limit specified by the maxRows parameter. To specify no limit on the number of rows to retrieve, set the maxRows parameter to -1.
	The database context must have been set by calling the appropriate method, such as <code>createBillingContext()</code> for billing entity objects. Subclasses of the IBXEntity class must implement this method because there is no default implementation. Returns a vector that contains the selected records. The following example retrieves all documents whose container ID field matches the object ID of this container. <pre>BXFilter filter = new BXFilter(); filter.addCondition("containerID", BXFilter.OP_EQUAL, getPrimaryKey()); Vector result = document.ejbFindAll(filter, null);</pre>

ejbFindByPrimaryKey()

Retrieves a record, identified by its key, from a data repository.

Syntax	<code>void ejbFindByPrimaryKey(String key) throws Exception;</code>
Parameters	<i>key</i> — A string that specifies the object ID
Discussion	The <i>key</i> parameter specifies the record to read. The fields of this object are set by from the Contents of the record. The database context must have been set by calling the appropriate method, such as <code>createBillingContext()</code> for billing entity objects. Subclasses of the IBXEntity class must implement this method because there is no default implementation.

ejbLoad()

Retrieves a record, identified by object ID, from a data repository.

Syntax	<code>void ejbLoad() throws Exception;</code>
Parameters	None.
Discussion	The primary key specifies the record to read. The fields of this object are set from the Contents of the record. The database context must be set by calling the appropriate method, such as <code>createBillingContext()</code> for billing entity objects. Subclasses of the IBXEntity class must implement this method because there is no default implementation.

ejbRemove()

Removes a record from a data repository.

Syntax	<code>void ejbRemove() throws Exception;</code>
Parameters	None.
Discussion	The primary key specifies the record to remove. The fields of the object itself are not changed. The database context must have been set by calling the appropriate method, such as <code>createBillingContext()</code> for billing entity objects. Subclasses of the IBXEntity class must implement this method because there is no default implementation.

ejbStore()

Updates a record in a data repository.

Syntax	<code>void ejbStore() throws Exception;</code>
Parameters	None.
Discussion	The primary key specifies the record to update. The database context must have been set by calling the appropriate method, such as <code>createBillingContext()</code> for billing entity objects. Subclasses of the IBXEntity class must implement this method; there is no default implementation.

NOTE	You must retrieve the record before you update it; otherwise, fields that you do not explicitly set will be written as null.

getName()

Determines the class name.

Syntax	<code>String getName() throws Exception;</code>
Parameters	None.
Discussion	Returns a string that contains the name of the entity's class. Subclasses of the IBXEntity class must implement this method because there is no default implementation. Each subclass implementation must return the name of the subclass. The following example implements the <code>getName()</code> method for the <code>BXDocument</code> class: <pre>public String getName() throws Exception { return ("BXDocument");}</pre>

getPrimaryKey()

Determines the primary key associated with a record.

Syntax	<code>String getPrimaryKey();</code>

Parameters	None.
Discussion	Returns a string that contains the primary key used to access a record.

getStatus()

Determines the status object currently associated with this entity.

Syntax	<code>BXStatus getStatus() ;</code>
Parameters	None.
Discussion	Returns a BXStatus object that specifies the current status.

getTime()

Determines the elapsed time since the object was created.

Syntax	<code>long getTime() ;</code>
Parameters	None.
Discussion	Returns a long integer that contains the number of milliseconds since the object was created.

setPrimaryKey()

Specifies the primary key associated with a record.

Syntax	<code>void setPrimaryKey(String objID) ;</code>
Parameters	<i>objID</i> — A string that specifies the object ID.
Discussion	Sets the key that is used by the <code>ejbLoad()</code> method to specify the record to read.

setStatus()

Specifies the status object to associate with this entity.

Syntax	<code>void setStatus (BXStatus status) ;</code>
Parameters	<i>status</i> — A BXStatus object that specifies the status
Discussion	

SearchResultSet Class Reference

The SearchResultSet class returns the result set from a JDBC lookup.

Class Definition

Table 1-71 describes the SearchResultSet class definition.

Table 1-71 SearchResultSet Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.common;</code>
Syntax	<code>public class SearchResultSet {...}</code>

Constructor

The SearchResultSet Class contains the following constructor.

Syntax	<code>public SearchResultSet();</code>
Parameters	None.
Discussion	Creates a SearchResultSet class object.

Fields

The SearchResultClass contains the following fields, listed in Table 1-72.

Table 1-72 SearchResultSet Fields

Field	Description
<code>public Vector resultSet;</code>	Result set from a JDBC lookup.
<code>int total;</code>	Total items in a result set.

Billing Classes Component Business Object Interfaces

This chapter discusses component business object interface classes for billing records. These classes provide access to records stored in an Oracle database.

This chapter includes the following sections:

- Overview
- Billing Class Reference Descriptions

Overview

Billing classes represent billing records in the database. You use these classes to manipulate these records. Table 2-1 lists the classes.

Table 2-1 Billing Classes (1 of 3)

Class Name	Description	Location
1. BXBillerCustAcct	Base class that manipulates BillerXpert customer account objects.	page 139
2. IBXBillerCustAcct	Interface that manipulates BillerXpert customer account objects.	page 139
3. BXBillingContext	Communicate between an AppLogic and an EJB component.	page 162
4. BXBillingEntity	Base class for billing business objects.	page 164
5. IBXBillingEntity	Interface for billing business objects.	page 167

Table 2-1 Billing Classes (2 of 3)

Class Name	Description	Location
6. BXBillingMgr	Creates billing-related component business objects.	page 167
7. BXCardInfo	Base class for credit card information.	page 167
8. IBXCardInfo	Interface for credit card information.	page 172
9. BXContainer	Base class that manipulates container objects.	page 184
10. IBXContainer	Interface that manipulates container objects.	page 184
11. BXDetail	Base class that manipulates detail line objects	page 194
12. IBXDetail	Interface that manipulates detail line objects	page 194
13. BXDocument	Base class that manipulates billing objects.	page 197
14. IBXDocument	Interface that manipulates billing objects.	page 198
15. BXEventLog	Base class that manipulates event log objects.	page 215
16. IBXEventLog	Interface that manipulates event log objects.	page 215
17. BXEventLogDetail	Manipulates event log detail objects.	page 221
18. BXExtensionDesc	Base class that represents the extension description table.	page 223
19. IBXExtensionDesc	Interface that represents the extension description table.	page 223
20. BXLineItem	Base class that manipulates objects that relate to purchase order line items.	page 225
21. IBXLineItem	Interface that manipulates objects that relate to purchase order line items.	page 226
22. BXMessage	Base class that manipulates message objects.	page 239
23. IBXMessage	Interface that manipulates message objects.	page 239
24. BXPAYHistory	Manipulates payment history objects.	page 251
25. IBXPAYHistory	Manipulates payment history objects.	page 253
26. BXPAYSchedule	Base class that manipulates payment schedule objects.	page 267
27. IBXPAYSchedule	Interface that manipulates payment schedule objects.	page 281
28. BXSUMMARY	Base class that manipulates billing summary objects.	page 282

Table 2-1 Billing Classes (3 of 3)

Class Name	Description	Location
29. IBXSummary	Interface that manipulates billing summary objects.	page 282

BXBillerCustAcct Class Reference

The BXBillerCustAcct class is the base class for defining methods that manipulate objects representing customer account records. It also defines methods that allow access to the account and related records.

Whenever possible, use the interface class IBXBillerCustAcct (see “IBXBillerCustAcct Class Reference) instead of BXBillerCustAcct.

IBXBillerCustAcct Class Reference

The IBXBillerCustAcct interface class manipulates BillerXpert B2B Edition customer account objects.

Class Definition

Table 2-2 describes the IBXBillerCustAcct interface definition.

Table 2-2 IBXBillerCustAcct Interface Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public interface IBXBillerCustAcct extends IBXBillingEntity {...}</code>

Methods

This section describes the IBXBillerCustAcct class methods. Table 2-3 lists these methods.

Before you can call the `IBXEntity.ejbCreate()` (see “IBXBillingEntity Class Reference,” on page 164) method to create a new record associated with the object, you must call the `setBillerID()`, `setUserID()`, and `setAccountNumber()` methods to set the biller ID, user ID, and account number fields, respectively.

Table 2-3 IBXBillerCustAcct Methods (1 of 3)

Methods	Description
Manipulating customer account and related records	
<code>ejbAssociateUser()</code>	Retrieves a customer account record by account name.
<code>ejbFindAllDocuments()</code>	Retrieves documents associated with a customer account.
<code>ejbFindAllContainers()</code>	Retrieves containers associated with a customer account.
<code>ejbFindAllBCAs()</code>	Retrieves all customer accounts registered to a user.
<code>ejbFindAllUserIDs()</code>	Retrieves all user IDs associated with a customer account.
<code>ejbFindPaymentSchedule()</code>	Retrieves the payment schedule for a customer account.
<code>ejbFindByAccountNumber()</code>	Associates an account with a user.
<code>ejbRemoveAssociatedUser()</code>	Disassociates an account from a user.
<code>ejbFindAllPaymentHistories()</code>	Retrieves payment history for a customer account
<code>ejbFindAllAutoPaySchedules</code>	Retrieves autopay schedules for a customer account
Getting and setting customer account information	
<code>getBillerID()</code>	Retrieves the biller ID.
<code>setBillerID()</code>	Specifies the biller ID.
<code>getAccountType()</code>	Retrieves the customer’s account type.
<code>setAccountType()</code>	Specifies the customer’s account type.
<code>getCustomerType()</code>	Retrieves the customer type.
<code>setCustomerType()</code>	Specifies the customer type.
<code>getDescription()</code>	Retrieves the customer’s description.
<code>setDescription()</code>	Specifies the customer’s description.
<code>getTaxID()</code>	Retrieves the customer’s tax ID.
<code>setTaxID()</code>	Specifies the customer’s tax ID.
<code>getSince()</code>	Retrieves the year in which the account was established.
<code>setSince()</code>	Specifies the year in which the account was established.

Table 2-3 IBXBillerCustAcct Methods (2 of 3)

Methods	Description
<code>getServiceOptions()</code>	Retrieves the account's service options.
<code>setServiceOptions()</code>	Specifies the account's service options.
<code>getProfile()</code>	Retrieves the profile.
<code>setProfile()</code>	Specifies the profile.
<code>getAccountNumber()</code>	Retrieves the customer's account number.
<code>setAccountNumber()</code>	Specifies the customer's account number.
<code>getAccountStatus()</code>	Retrieves the account status.
<code>setAccountStatus()</code>	Specifies the account status.
<code>getUnpaidBills()</code>	Retrieves the quantity of unpaid bills.
<code>setUnpaidBills()</code>	Specifies the quantity of unpaid bills.
<code>getUnpaidBalance()</code>	Retrieves the unpaid balance.
<code>setUnpaidBalance()</code>	Specifies the unpaid balance.
<code>getAccountBalance()</code>	Retrieves the account balance.
<code>setAccountBalance()</code>	Specifies the account balance.
<code>getAccountBalanceDate()</code>	Retrieves the balance due date.
<code>setAccountBalanceDate()</code>	Specifies the balance due date.
<code>getCreateDate()</code>	Retrieves the customer account's creation date.
<code>getModifyDate()</code>	Retrieves the customer account's modification date.
<code>getDisputeStatus()</code>	Retrieves the dispute status.
<code>setDisputeStatus()</code>	Specifies the dispute status.
<code>getExtension()</code>	Retrieves the customer's extension.
<code>setExtension()</code>	Specifies the customer's extension.
<code>getUserID()</code>	Retrieves the user ID.
<code>setUserID()</code>	Specifies the user ID.
<code>getBillingAddress()</code>	Retrieves the customer's billing address.
<code>setBillingAddress()</code>	Specifies the customer's billing address.
<code>getServiceAddress()</code>	Retrieves the customer's service address ID.
<code>setServiceAddress()</code>	Specifies the customer's service address ID.
<code>getVerifyUser()</code>	Retrieves the user.

Table 2-3 IBXBillerCustAcct Methods (3 of 3)

Methods	Description
<code>setVerifyUser()</code>	Specifies the user.
<code>getAuthorizedUser()</code>	Retrieves the authorized user.
<code>setAuthorizedUser()</code>	Specifies the authorized user.
<code>getAuthorizedDate()</code>	Retrieves the authorized date.
<code>setAuthorizedDate()</code>	Specifies the authorized date.
<code>getVerifiedDate()</code>	Retrieves the verification date.
<code>setVerifiedDate</code>	Specifies the verification date.
<code>getFileID()</code>	Retrieves the file ID.
<code>setFileID()</code>	Specifies the file ID.
<code>getAutopay()</code>	Retrieves the autopay flag.
<code>setAutopay()</code>	Specifies the autopay flag.
<code>getExtDate1</code>	Retrieves the first extension date.
<code>setExtDate1</code>	Specifies the first extension date.
<code>getExtDate2()</code>	Retrieves the second extension date.
<code>setExtDate2()</code>	Specifies the second extension date.
<code>getExtDate3()</code>	Retrieves the third extension date
<code>setExtDate3()</code>	Specifies the third extension date.
<code>getExtDate4()</code>	Retrieves the fourth extension date
<code>setExtDate4()</code>	Specifies the fourth extension date.
<code>getExtNum1()</code>	Retrieves the first extension number.
<code>setExtNum1()</code>	Specifies the first extension number.
<code>getExtNum2()</code>	Retrieves the second extension number.
<code>setExtNum2()</code>	Specifies the second extension number.
<code>getExtNum3()</code>	Retrieves the third extension number.
<code>setExtNum3()</code>	Specifies the third extension number.
<code>getExtNum4()</code>	Retrieves the fourth extension number.
<code>setExtNum4()</code>	Specifies the fourth extension number.
<code>getCompanyID()</code>	Determines Company ID.
<code>setCompanyID()</code>	Specifies Company ID.

ejbAssociateUser()

Associates an account with a user.

Syntax	<code>public void ejbAssociateUser(String acctNo, String userID, String companyID) throws Exception;</code>
Parameters	<i>acctNo</i> — A string that specifies the account number. <i>userID</i> — A string that specifies the user ID. <i>companyID</i> — String that specifies the company ID.
Discussion	Retrieves the customer account record identified by the account number from the database and specifies the fields of this object from the contents of the record. It then associates the specified account number with the specified user and updates the association in the database. The customer account's database context must previously have been set by calling this object's <code>createBillingContext()</code> method. The customer account record identified by the account number must already exist in the database. The <code>ejbAssociateUser()</code> method can be committed or rolled back as part of a transaction.

ejbFindAllBCAs()

Retrieves all customer accounts registered to a user.

Syntax	<code>public Vector ejbFindAllBCAs(String companyID) throws Exception;</code> <code>public Vector ejbFindAllBCAs(String userID, String companyID) throws Exception;</code>
Parameters	<i>userID</i> — String that specifies the user ID. <i>companyID</i> — String that specifies the company ID.
Discussion	Retrieves all customer account records for the customer specified by user ID. The customer account object's database context must previously have been set by calling each object's <code>createBillingContext()</code> method. For each record, the <code>ejbFindAllBCA()</code> method creates an object, specifies the fields in the object from a retrieved record, and adds the object to the return vector. Returns a vector that contains <code>BXBillerCustAcct</code> objects representing the selected records.

ejbFindAllContainers()

Retrieves containers associated with a customer account.

Syntax	<code>public Vector ejbFindAllContainers(BxFilter <i>filter</i>, BxOrder <i>order</i>) throws Exception;</code>
Parameters	<i>filter</i> — BxFilter object that specifies the Where clause of an SQL Select statement. <i>order</i> — BxOrder that specifies the Order By clause of an SQL Select statement
Discussion	The <code>ejbFindAllContainers()</code> method retrieves all container records for the account that match the clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, the <code>ejbFindAllContainers()</code> method creates an object, specifies the fields in the object from the record, and adds the object to the return vector. If the <i>filter</i> parameter is null, all records will be retrieved. If the <i>order</i> parameter is null, the retrieved records will be sorted in ascending order by their primary key. The customer account object's database context must previously have been set by calling each object's <code>createBillingContext()</code> method. The account ID must be set in the customer account object.

ejbFindAllDocuments()

Retrieves all documents associated with a customer account.

Syntax	<code>public Vector ejbFindAllDocuments(BxFilter <i>filter</i>, BxOrder <i>order</i>) throws Exception;</code>
Parameters	<i>filter</i> — BxFilter object that specifies the Where clause of an SQL Select statement. If <i>filter</i> is null, all records will be retrieved. <i>order</i> — BxOrder that specifies the Order By clause of an SQL Select statement. If <i>order</i> is null, the retrieved records are sorted in ascending order by their primary key.

Discussion	Retrieves all document records for an account that match the clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, <code>ejbFindAllDocuments()</code> creates an object, specifies the fields in the object from the record, and adds the object to the return vector. The customer account object's database context must previously have been set by calling each object's <code>createBillingContext()</code> method. The account ID must be set in the customer account object. Returns a vector that contains <code>BXDocument</code> objects representing the selected records.

ejbFindAllUserIDs()

Retrieves all user IDs associated with a customer account.

Syntax	<code>public Vector ejbFindAllUserIDs(String acctNo) throws Exception;</code>
Parameters	<i>acctNo</i> — String that specifies the account number
Discussion	Returns a vector that contains strings, each of which is a user ID. The customer account's database context must previously have been set by calling this object's <code>createBillingContext()</code> method.

ejbFindByAccountNumber()

Retrieves a customer account record by account name.

Syntax	<code>public void ejbFindByAccountNumber(String acctNo, String biller)throws Exception;</code>
Parameters	<i>acctNo</i> — A string that specifies the account number. <i>biller</i> — A string that specifies the account name
Discussion	Retrieves the customer account record specified by the account number. The object's fields are set from the contents of the record. The customer account's database context must previously have been set by calling this object's <code>createBillingContext()</code> method. The values of fields already set in the object are overwritten by this method; values in the database are not affected.

ejbFindPaymentSchedule()

Retrieves the payment schedule for a customer account.

Syntax	<pre>public IBXPaySchedule ejbFindPaymentSchedule() throws Exception</pre>
Parameters	None.
Discussion	Returns the IBXPaySchedule object associated with this customer account or null if no IBXPaySchedule object is associated with the account. The customer account object's database context must previously have been set by calling each object's createBillingContext () method. The account ID must be set in the customer account object.

ejbRemoveAssociatedUser()

Disassociates an account from a user.

Syntax	<pre>public void ejbRemoveAssociatedUser(String acctNo, String userID) throws Exception;</pre>
Parameters	<i>acctNo</i> — A string that specifies the account number <i>userID</i> — A string that specifies the user ID
Discussion	Removes the association between the specified account number and the specified user from the database. The association must already exist in the database. The customer account's database context must previously have been set by calling this object's createBillingContext () method. The ejbAssociateUser () method can be committed or rolled back as part of a transaction

ejbFindAllPaymentHistories()

Retrieves payment history for a customer account.

Syntax	<pre>public Vector ejbFindAllPaymentHistories (BXfilter filter, BXOrder order) throws Exception;</pre>
Parameters	<i>filter</i> — Parameter for selecting the appropriate history <i>order</i> — Sequence in payment history

Discussion	Finds the association between the specified account number and the specified user from the database. The association must already exist in the database. The customer account's database context must previously have been set by calling this object's <code>createBillingContext()</code> method. The <code>ejbAssociateUser()</code> (see page 143) method can be committed or rolled back as part of a transaction.

ejbFindAllAutoPaySchedules()

Retrieves autopay schedules for a customer account.

Syntax	<code>public Vector ejbFindAllPaymentHistories (BXfilter <i>filter</i>, BXOrder <i>order</i>) throws Exception;</code>
Parameters	<i>filter</i> — A parameter for selecting the appropriate history. <i>order</i> — The sequence in payment scheduling.
Discussion	Finds the association between the specified account number and the specified user from the database. The association must already exist in the database. The customer account's database context must previously have been set by calling this object's <code>createBillingContext()</code> method. The <code>ejbAssociateUser()</code> method can be committed or rolled back as part of a transaction

getAccountBalance()

Retrieves the account balance.

Syntax	<code>public double getAccountBalance();</code>
Parameters	None.
Discussion	Returns a double-precision floating number that contains the balance.

getAccountBalanceDate()

Retrieves the balance due date.

Syntax	<code>public Date getAccountBalanceDate();</code>
Parameters	None.
Discussion	Returns the <code>Date</code> object that contains the date

getAccountNumber()

Retrieves the customer’s account number.

Syntax	<code>public String getAccountNumber();</code>
Parameters	None.
Discussion	Returns a string that contains the account number.

getAccountStatus()

Retrieves the account status.

Syntax	<code>public String getAccountStatus();</code>
Parameters	None.
Discussion	Returns a string that contains the account status.

getAccountType()

Retrieves the customer’s account type.

Syntax	<code>public String getAccountType();</code>
Parameters	None.
Discussion	Returns a string that contains the account type.

getBillerID()

Retrieves the biller ID.

Syntax	<code>public String getBillerID();</code>
Parameters	None.
Discussion	Returns a string that contains the ID.

getBillingAddress()

Retrieves the customer’s billing address.

Syntax	<code>public String getBillingAddressID();</code>

Parameters	None.
Discussion	Returns a string that contains the address.

getCustomerType()

Retrieves the customer's account type.

Syntax	<code>public String getCustomerType();</code>
Parameters	None.
Discussion	Returns a string that contains the customer type.

getDescription()

Retrieves the customer's description.

Syntax	<code>public String getDescription();</code>
Parameters	None.
Discussion	Returns a string that contains the customer description.

getDisputeStatus()

Retrieves the dispute status.

Contents	Description
Syntax	<code>public String getDisputeStatus();</code>
Parameters	None.
Discussion	Returns a string that contains the status.

getExtension()

Retrieves the customer’s extension.

Contents	Description
Syntax	<code>public String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the extension.

getModifyDate()

Retrieves the customer account’s modification date.

Syntax	<code>public Date getModifyDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the modification date.

getProfile()

Retrieves the profile.

Syntax	<code>public String getProfile();</code>
Parameters	None.
Discussion	Returns a string that contains the profile.

getServiceAddressID()

Retrieves the customer’s service address ID.

Syntax	<code>public String getServiceAddressID();</code>
Parameters	None.
Discussion	Returns a string that contains the address.

getServiceOptions()

Retrieves the account's service options.

Syntax	<code>public String getServiceOptions();</code>
Parameters	None.
Discussion	Returns a string that contains the service options.

getSince()

Retrieves the year in which the account was established.

Syntax	<code>public int getSince();</code>
Parameters	None.
Discussion	Returns an integer that contains the year.

getTaxID()

Retrieves the customer's tax ID.

Syntax	<code>public String getTaxID();</code>
Parameters	None.
Discussion	Returns a Date object that contains the tax identification.

getUnpaidBalance()

Retrieves the unpaid balance.

Syntax	<code>public double getUnpaidBalance();</code>
Parameters	None.
Discussion	Returns a double-precision floating number that contains the balance.

getUnpaidBills()

Retrieves the quantity of unpaid bills.

Syntax	<code>public int getUnpaidBills();</code>

Parameters	None.
Discussion	Returns an integer that contains the quantity.

getUserID()
Retrieves the user ID.

Syntax	<code>public String getUserID();</code>
Parameters	None.
Discussion	Returns a string that contains the ID.

getVerifiedUser()
Retrieves the verification date.

Syntax	<code>public String getVerifiedUser();</code>
Parameters	None.
Discussion	Returns a string that verified user.

setAuthorizedUser()
Retrieves the authorization of an user.

Syntax	<code>public String getAuthorizedUser();</code>
Parameters	None.
Discussion	Returns a string that authorizes the user.

getAuthorizedDate()
Retrieves the authorization date.

Syntax	<code>public String getAuthorizeDate();</code>
Parameters	None.
Discussion	Returns a string that authorizes the date.

getVerifiedDate()

Retrieves the verification date.

Syntax	<code>public String getVerifiedDate();</code>
Parameters	None.
Discussion	Returns a string that verifies the authorization date.

getFileID()

Retrieves the file ID.

Syntax	<code>public String getFileID();</code>
Parameters	None.
Discussion	Returns a string that verified the fileID.

getAutoPay()

Retrieves the autopay amount.

Syntax	<code>public String getAutoPay();</code>
Parameters	None.
Discussion	Returns a string that verifies the autopay amount.

setAccountBalance()

Specifies the account balance.

Syntax	<code>public void setAccountBalance(double <i>acctBalance</i>);</code>
Parameters	<i>acctBalance</i> — A double-precision floating point number that specifies the balance
Discussion	Only changes the value of the account balance in the object; it does not modify the value in the database.

setAccountBalanceDate()

Specifies the account balance due date.

Syntax	<code>public void setAccountBalanceDate(Date <i>acctBalanceDate</i>) ;</code>
Parameters	<i>acctBalanceDate</i> — A Date object that specifies the date the account balance is due.
Discussion	Only changes the value of the account balance date in the object; it does not modify the value in the database.

setAccountNumber()

Specifies the customer's account number.

Syntax	<code>public void setAccountNumber(String <i>acctNum</i>) ;</code>
Parameters	<i>acctNumber</i> — A string that specifies the account number
Discussion	Only specifies of the account number in the object; it does not modify the value in the database.

setAccountStatus()

Specifies the account status.

Syntax	<code>public void setAccountStatus(String <i>acctStatus</i>) ;</code>
Parameters	<i>acctStatus</i> — A string that specifies the account status.
Discussion	Only specifies of the account status in the object; it does not modify the value in the database.

setAccountType()

Specifies the customer's account type.

Syntax	<code>public void setAccountType(String <i>acctType</i>) ;</code>
Parameters	<i>acctType</i> — Specifies the account type
Discussion	Only specifies of the account type in the object; it does not modify the value in the database.

setBillerID()

Specifies the biller ID.

Syntax	<code>public void setBillerID(String <i>billerID</i>);</code>
Parameters	<i>billerID</i> — A string that specifies the biller ID
Discussion	Only specifies of the biller ID in the object; it does not modify the value in the database.

setBillingAddress()

Specifies the customer's billing address.

Syntax	<code>public void setBillingAddressID (String <i>addressID</i>);</code>
Parameters	<i>addressID</i> — A string that specifies the biller address ID
Discussion	Only specifies of the biller address ID in the object; it does not modify the value in the database.

setCustomerType()

Specifies the customer type.

Syntax	<code>public void setCustomerType(String <i>CustomerType</i>);</code>
Parameters	<i>customerType</i> — String that specifies the customer type
Discussion	Only specifies of the customer type in the object; it does not modify the value in the database.

setDescription()

Specifies the customer's description.

Syntax	<code>public void setDescription(String <i>description</i>);</code>
Parameters	<i>description</i> — A string that specifies the description
Discussion	Only specifies of the description of the object; it does not modify the value in the database.

setDisputeStatus()

Specifies the dispute status.

Syntax	<code>public void setDisputeStatus(String <i>disputeStatus</i>);</code>
Parameters	<i>disputeStatus</i> — String that specifies the dispute status
Discussion	Only specifies of the dispute status of the object; it does not modify the value in the database.

setExtension()

Specifies the customer’s extension.

Syntax	<code>public void setExtension (String <i>extension</i>);</code>
Parameters	<i>extension</i> — String that specifies the extension
Discussion	Only specifies of the extension of the object; it does not modify the value in the database.

setProfile()

Specifies the profile.

Syntax	<code>public void setProfile(String <i>profile</i>);</code>
Parameters	<i>profile</i> — String that specifies the profile.
Discussion	Only specifies of the profile of the object; it does not modify the value in the database.

setServiceAddressID()

Specifies the customer’s service address ID.

Syntax	<code>public void setServiceAddressID (String <i>addressID</i>);</code>
Parameters	<i>addressID</i> — String that specifies the address ID
Discussion	Only specifies of the service ID in the object; it does not modify the value in the database.

setServiceOptions()

Specifies the account's service options.

Syntax	<code>public void setServiceOptions(String <i>serviceOption</i>);</code>
Parameters	<i>serviceOption</i> — A string that specifies the service option
Discussion	Only specifies of the service option in the object; it does not modify the value in the database.

setSince()

Specifies the year in which the account was established.

Syntax	<code>public void setSince (int <i>Since</i>);</code>
Parameters	<i>since</i> — An integer that specifies the year.
Discussion	Only specifies of the value change field in the object; it does not modify the value in the database.

setTaxID()

Specifies the customer's tax ID.

Syntax	<code>public void setTaxID(String <i>taxID</i>);</code>
Parameters	<i>taxID</i> — A string that specifies the tax ID
Discussion	Only specifies of the tax ID in the object; it does not modify the value in the database.

setUnpaidBalance()

Specifies the unpaid balance.

Syntax	<code>public void setUnpaidBalance (double <i>unpaidBalance</i>);</code>
Parameters	<i>unpaidBalance</i> — String that specifies the unpaid balance
Discussion	Only specifies of the unpaid balance in the object; it does not modify the value in the database.

setUnpaidBills()

Specifies the quantity of unpaid bills.

Syntax	<code>public void setUnpaidBills(int <i>unpaidBills</i>);</code>
Parameters	<i>unpaidBills</i> — String that specifies the unpaid balance
Discussion	Only specifies of the unpaid balance in the object; it does not modify the value in the database.

setUserID()

Specifies the user ID.

Syntax	<code>public void setUserID(String userID);</code>
Parameters	<i>userID</i> — A string that specifies the user ID
Discussion	Only specifies of the user ID in the object; it does not modify the value in the database.

getExtDate1()

Retrieves the first extension date.

Syntax	<code>public Date getExtDate1();</code>
Parameters	None.
Discussion	

setExtDate1()

Specifies the credit card first extension date.

Syntax	<code>public void setExtDate1 (Date <i>date</i>);</code>
Parameters	<i>date</i> — The first extension date.
Discussion	

getExtDate2()

Retrieves the second extension date.

Syntax	<code>public Date getExtDate2();</code>
Parameters	None.
Discussion	

setExtDate2()

Specifies the second extension date.

Syntax	<code>public void setExtDate1 (Date <i>date</i>);</code>
Parameters	<i>date</i> — The second extension date.
Discussion	

getExtDate3()

Retrieves the third extension date.

Syntax	<code>public Date getExtDate3();</code>
Parameters	None.
Discussion	

setExtDate1()

Specifies the third extension date.

Syntax	<code>public void setExtDate3 (Date <i>date</i>);</code>
Parameters	<i>date</i> — The third extension date.
Discussion	

getExtDate4()

Retrieves the fourth extension date.

Syntax	<code>public Date getExtDate4();</code>

Parameters	None.
Discussion	

setExtDate4()
Specifies the fourth extension date.

Syntax	<code>public void setExtDate4 (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — The fourth extension date.
Discussion	

getExtNum1()
Retrieves the first extension number.

Syntax	<code>public double getExtNum1 () ;</code>
Parameters	None.
Discussion	

setExtNum1()
Specifies the first extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>) ;</code>
Parameters	<i>num</i> — The first extension number.
Discussion	

getExtNum2()
Retrieves the second extension number.

Syntax	<code>public double getExtNum2 () ;</code>
Parameters	None.
Discussion	

setExtNum2()

Specifies the second extension number.

Syntax	<code>public void setExtNum2 (double <i>num</i>);</code>
Parameters	<i>num</i> — The second extension number.
Discussion	

getExtNum3()

Retrieves the third extension number.

Syntax	<code>public double getExtNum3();</code>
Parameters	None.
Discussion	

setExtNum3()

Specifies the third extension number.

Syntax	<code>public void setExtNum3 (double <i>num</i>);</code>
Parameters	<i>num</i> — The third extension number.
Discussion	

getExtNum4()

Retrieves the fourth extension number.

Syntax	<code>public double getExtNum4();</code>
Parameters	None.
Discussion	

getCompanyID()

Determines the Company ID.

Syntax	<code>public String getCompanyID();</code>
Parameters	None.
Discussion	Returns the Company ID associated with the current context

setCompanyID()

Specifies the Company ID.

Syntax	<code>public void setCompanyID (String <i>companyID</i>);</code>
Parameters	<i>CompanyID</i> — A string that specifies the company ID.
Discussion	Returns a reference to the status object.

BXBillingContext Class Reference

The BXBillingContext class is used to associate database related parameters or information with the billing EJBs. BXBillingContext extends the BXContext class, which is accessed via the IBXDBCContext interface (see “IBXDBCContext Class Reference,” on page 72).

Instances of the BXBillingContext class represent a database context for billing objects. Typically you create a billing object using one of the BXBillingMgr class methods (see “BXBillingMgr Class Reference,” on page 167) and then create a context for it by creating a BXBillingContext object.

Class Definition

Table 2-2 describes the BXBillingContext class definition.

Table 2-4 BXBillingContext Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public class BXBillingContext extends BXDBCContext {...}</code>

Constructor

This section describes the BXBillingContext constructor. Table 2-5 lists this constructor.

Table 2-5 BXBillingContext Constructors

Constructor	Description
BXBillingContext()	Creates a BXBillingContext object. (Note: there are four instances of the BXBillingContext constructors.)

BXBillingContext()

This constructor creates a BXBillingContext object.

Syntax	<pre>public BXBillingContext (IContext context, String billerID) throws Exception; public BXBillingContext (IContext context, BXStatus status, String billerId) throws Exception; public BXBillingContext (IContext context, BXStatus status, String billerId, String companyId) throws Exception; public BXBillingContext (IContext context, String datasource, String database, String username, String password, int trace);</pre>
Parameters	<i>context</i> — An IContext object that specifies the context. <i>status</i> — A BXStatus object that holds the current status of the context. <i>datasource</i> — A string that specifies the data source. <i>database</i> — A string that specifies the database. <i>username</i> — A string that specifies the user name. <i>password</i> — A string that specifies the password. <i>billerId</i> — A string that specifies the biller id. <i>companyId</i> — A string that specifies the company id. <i>trace</i> — An integer that specifies the debug trace level.

Discussion	You typically create a billing context object by specifying the applogic's <i>context</i> variable for the context parameter. The first and third forms of the constructor do not associate a status object with the context. In this case you, call the <code>setStatus()</code> or <code>getBXStatus()</code> method to specify a status object for the context. The first two forms of the constructor take the database login information from a configuration file. If you use the third form of the constructor, specify the same value for the <code>datasource</code> parameter as you do for the database parameter.

BXBillingEntity Class Reference

The BXBillingEntity class is the base class for billing business objects.

Whenever possible, use the interface class IBXBillingEntity (see “IBXBillingEntity Class Reference) instead of BXBillingEntity.

IBXBillingEntity Class Reference

The IBXBillingEntity class is the interface for the Billing Entity bean for all billing classes. You must call the `setBillingContext()` method of each object that represents a billing record in the database before you can manipulate the record with the objects.

Class Definition

Table 2-6 describes the IBXBillingEntity interface definition.

Table 2-6	BXBillingEntity Interface Definition
Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public interface IBXBillingEntity extends IBXEntity {...}</code>

Methods

This section describes the IBXBillingEntity interface methods. Table 2-7 lists these methods.

Table 2-7 IBXBillingEntity Interface Methods

Methods	Description
Getting and setting billing context information	
<code>getBillingContext()</code>	Retrieves the billing context.
<code>setBillingContext()</code>	Specifies the billing context.
<code>getExtension1()</code>	Retrieves the extension1 context.
<code>setExtension1()</code>	Specifies the extension1 context.
<code>getExtension2()</code>	Retrieves the extension2 context.
<code>setExtension2()</code>	Specifies the extension2 context.
<code>getExtension3()</code>	Retrieves the extension3 context.
<code>setExtension3()</code>	Specifies the extension3 context.
<code>getExtension4()</code>	Retrieves the extension4 context.
<code>setExtension4()</code>	Specifies the extension4 context.

`getBillingContext()`

Retrieves the billing context.

Syntax	<code>public BXBillingContext getBillingContext();</code>
Parameters	None.
Discussion	Returns the BXBillingContext object associated with this billing object.

`getExtension1()`

Retrieves the first extension context.

Contents	Description
Syntax	<code>public String getExtension1();</code>
Parameters	None.
Discussion	Returns the BXExtension1 object.

getExtension2()

Retrieves the second extension context.

Syntax	<code>public String getExtension2();</code>
Parameters	None.
Discussion	Returns the <code>BXExtension2</code> object.

getExtension3()

Retrieves the third extension context.

Syntax	<code>public String getExtension3();</code>
Parameters	None.
Returns	The <code>BXExtension3</code> object.

getExtension4()

Retrieves the fourth extension context.

Syntax	<code>public String getExtension4();</code>
Parameters	None.
Discussion	Returns the <code>BXExtension4</code> object.

setBillingContext()

Specifies the billing context.

Syntax	<code>public void setBillingContext (<i>BXBillingContext context</i>);</code>
Parameters	<i>context</i> — The context associated with this billing object.
Discussion	

setExtension1()

Specifies the first extension context.

Syntax	<code>public void setExtension1(String <i>value</i>);</code>

Parameters	<i>value</i> — The context associated with the first extension object.
Discussion	

setExtension2()
Specifies the second extension context.

Contents	Description
Syntax	<code>public void setExtension2(String <i>value</i>);</code>
Parameters	<i>value</i> — The context associated with the second extension object.
Discussion	

setExtension3()
Specifies the third extension context.

Syntax	<code>public void setExtension3 (String <i>value</i>);</code>
Parameters	<i>value</i> — The context associated with the third extension object.
Discussion	

setExtension4()
Specifies the fourth extension context.

Syntax	<code>public void setExtension4(String <i>value</i>);</code>
Parameters	<i>value</i> — The context associated with fourth extension object.
Discussion	

BXBillingMgr Class Reference

The BXBillingMgr class defines methods that create billing objects. You typically create a billing object and use it to manipulate a record in the database.

Class Definition

Table 2-8 describes the BXBillingMgr class definition.

Table 2-8 BXBillingMgr Class Definition	
Package	com.iplanet.ecomm.billxb2b.ejb.billing;
Syntax	public class BXBillingMgr {...}

Methods

This section describes the BXBillingMgr class methods. Table 2-8 lists these methods.

Table 2-9 BXBillingMgr Method Definitions

Methods	Description
Create billing objects	
createBillerCustAcct()	Creates a biller customer account object.
createContainer()	Creates a container object.
createDocument()	Creates a document object.
createSummary()	Creates a summary object.
createPaySchedule()	Creates a payment schedule object.
createPayHistory()	Creates a payment history object.
createEventLog()	Creates an event log object.
createMessage()	Creates a message object.
createCardInfo()	Creates a credit card object
Create entity object	
createObject()	Creates an object.

createBillerCustAcct()

Creates a biller customer account object.

Syntax	public static IBXBillerCustAcct createBillerCustAcct(String key) throws Exception;

Parameters	<i>key</i> — A string that contains the object mapping key in the format of <code>billerName.acctType</code> .
Discussion	The <code>createBillerCustAcct</code> method is a class method. If you use only the default account, specify null for the <code>billName acctType</code> . Returns an <code>IBXBillerCustAcct</code> object.

createCardInfo()

Creates a credit card object.

Syntax	<pre>public static IBXCardInfo createCardInfo (String key) throws Exception;</pre>
Parameters	<i>key</i> — A string that contains the object mapping key.
Discussion	Returns an <code>IBXCardInfo</code> object.

createContainer()

Creates a `BXContainer` object.

Syntax	<pre>public static IBXContainer createContainer (String key) throws Exception;</pre>
Parameters	<i>key</i> — A string that contains the object mapping key in the format of <code>billerName.acctType</code> .
Discussion	This method is a class method. If you use only the default account, specify null for the <code>billName acctType</code> . Returns an <code>IBXBillerContainer</code> object.

createDocument()

Creates a document object.

Syntax	<pre>public static IBXDocument createDocument (String key) throws Exception;</pre>
Parameters	<i>key</i> — A string that contains the object mapping key in the format of <code>billerName.acctType</code> .
Discussion	The <code>createDocument</code> method is a class method. If you use only the default account, specify null for the <code>billName acctType</code> . Returns an <code>IBXDetailDocument</code> object.

createEventLog()

Creates an event log object.

Syntax	<code>public static IBXEventLog createEventLog (String <i>key</i>) throws Exception;</code>
Parameters	<i>key</i> — A string that contains the object mapping key in the format of <code>billerName.acctType</code> .
Discussion	The <code>createEventLog</code> method is a class method. If you use only the default account, specify <code>null</code> for the <code>billName</code> <code>acctType</code> . Returns an <code>IBXEventLog</code> object.

createMessage()

Creates a message object.

Syntax	<code>public static IBXMessage createMessage(String <i>key</i>) throws Exception;</code>
Parameters	<i>key</i> — A string that contains the object mapping key in the format of <code>billerName.acctType</code> .
Discussion	The <code>createMessage</code> method is a class method. If you use only the default account, specify <code>null</code> for the <code>billName</code> <code>acctType</code> . Returns an <code>IBXMessage</code> object.

createObject()

Creates an object.

Syntax	<code>public static IBXEntity createObject (String <i>interfaceName</i>, String <i>key</i>) throws Exception;</code>
Parameters	<i>interfaceName</i> — A string that specifies the class to which the object belongs. <i>key</i> — A string that contains the object mapping key in the format of a <code>billerName.acct.Type</code> .
Discussion	The <code>createObject</code> method is a protected class method. If you use only the default account, specify <code>null</code> for the <code>billName</code> <code>acctType</code> . Returns an <code>IBXEntity</code> object.

createPayHistory()

Creates a payment history object.

Syntax	<pre>public static IBXPayHistory createPayHistory (String key) throws Exception;</pre>
Parameters	<i>key</i> — a string that contains the object mapping key in the format of a <code>billerName.acct.Type</code> .
Discussion	The <code>createPayHistory</code> is a class method. If you use only the default account, specify <code>null</code> for the <code>billName acctType</code> . Returns an <code>IBXPayHistory</code> object.

createPaySchedule()

Creates a payment schedule object.

Syntax	<pre>public static IBXPaySchedule createPaySchedule (String key) throws Exception;</pre>
Parameters	<i>key</i> — A string that contains the object mapping key in the format of a <code>billerName.acct.Type</code> .
Discussion	The <code>createPaySchedule</code> method is a class method. If you use only the default account, specify <code>null</code> for the <code>billName acctType</code> . Returns an <code>IBXPaySchedule</code> object.

createRequestLog()

Creates a request log object.

Syntax	<pre>static IOFXRequestLog createRequestLog (String key) throws Exception;</pre>
Parameters	<i>key</i> — A string that contains the object mapping key
Discussion	Returns an <code>IOFXRequestLog</code> object.

createSummary()

Creates a summary object.

Contents	Description
Syntax	<pre>public static IBXSummary createSummary (String key) throws Exception;</pre>

Contents	Description
Parameters	<i>key</i> — A string that contains the object mapping key in the format of a <code>billerName.acct.Type</code> .
Discussion	The <code>createSummary</code> method is a class method. If you use only the default account, specify <code>null</code> for the <i>billName</i> parameter. Returns an <code>IBXSummary</code> object.

BXCardInfo Class Reference

The `BXCardInfo` class is the base class for credit card information.

Whenever possible, use the interface class `IBXCardInfo` (see “`IBXCardInfo` Class Reference”) instead of `BSPBillerInfo`.

IBXCardInfo Class Reference

The `IBXCardInfo` class is the interface for credit card information.

Class Definition

Table 2-10 describes the `IBXCardInfo` class definition.

Table 2-10 `IBXCardInfo` Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public interface IBXCardInfo extends IBXBillingEntity;</code>

Methods

This section describes the `IBXCardInfo` class methods. Table 2-11 lists these methods.

Table 2-11 `IBXCardInfo` Methods (1 of 3)

Method	Description
<code>getType()</code>	Retrieves the card type (smart card or credit card).
<code>setType()</code>	Specifies the card type (smart card or credit card).
<code>getBrand()</code>	Retrieves the credit card type.

Table 2-11 IBXCardInfo Methods (2 of 3)

Method	Description
<code>setBrand()</code>	Specifies the credit card type
<code>getNumber()</code>	Retrieves the credit card number.
<code>setNumber()</code>	Specifies the credit card number.
<code>getDescription()</code>	Retrieves the credit card description.
<code>setDescription()</code>	Specifies the credit card description.
<code>getExpDate()</code>	Retrieves the credit card expiration date.
<code>setExpDate()</code>	Specifies the credit card expiration date.
<code>getOwner()</code>	Retrieves the credit card owner's name.
<code>setOwner()</code>	Specifies the credit card owner's name.
<code>getBillingName()</code>	Retrieves the credit card billing name.
<code>setBillingName()</code>	Specifies the credit card billing name.
<code>getAddress()</code>	Retrieves the credit card address.
<code>setAddress()</code>	Specifies the credit card address.
<code>getCity()</code>	Retrieves the credit card address city.
<code>setCity()</code>	Specifies the credit card address city.
<code>getState()</code>	Retrieves the credit card address state.
<code>setState()</code>	Specifies the credit card address state.
<code>getCountry()</code>	Retrieves the credit card address country.
<code>setCountry()</code>	Specifies the credit card address country.
<code>getPostalCode()</code>	Retrieves the credit card address postal code.
<code>setPostalCode()</code>	Specifies the credit card address postal code.
<code>getCardStatus()</code>	Retrieves the credit card address status.
<code>setCardStatus()</code>	Specifies the credit card address status.
<code>getPin()</code>	Retrieves the credit card address pin number.
<code>setPin()</code>	Specifies the credit card address pin number.
<code>getExtDate1()</code>	Retrieves the first extension date
<code>setExtDate1()</code>	Specifies the first extension date.
<code>getExtDate2()</code>	Retrieves the second extension date
<code>setExtDate2()</code>	Specifies the second extension date.

Table 2-11 IBXCardInfo Methods (3 of 3)

Method	Description
<code>getExtDate3 ()</code>	Retrieves the third extension date
<code>setExtDate3 ()</code>	Specifies the third extension date.
<code>getExtDate4 ()</code>	Retrieves the fourth extension date
<code>setExtDate4 ()</code>	Specifies the fourth extension date.
<code>getExtNum1 ()</code>	Retrieves the first extension number.
<code>setExtNum1 ()</code>	Specifies the first extension number.
<code>getExtNum2 ()</code>	Retrieves the second extension number.
<code>setExtNum2 ()</code>	Specifies the second extension number.
<code>getExtNum3 ()</code>	Retrieves the third extension number.
<code>setExtNum3 ()</code>	Specifies the third extension number.
<code>getExtNum4 ()</code>	Retrieves the fourth extension number.
<code>setExtNum4 ()</code>	Specifies the fourth extension number.

getType()

Retrieves the credit card or smart card type.

Syntax	<code>public String getType();</code>
Parameters	None.
Discussion	

setType()

Specifies the credit card or smart card type.

Syntax	<code>public void setType (String value);</code>
Parameters	<code>value</code> — The credit card or smart card type.
Discussion	

getBrand()

Retrieves the credit card type (Visa, MasterCard, AmEx, and so forth).

Syntax	<code>public String getBrand();</code>

Parameters	None.

Discussion

setBrand()

Retrieves the credit card type (Visa, MasterCard, Amex, and so forth).

Syntax	<code>public void setBrand (String value);</code>

Parameters	<i>value</i> — The credit card type.

Discussion

getNumber()

Retrieves the credit card number.

Syntax	<code>public String getNumber();</code>

Parameters	None.

Discussion

setNumber()

Specifies the credit card number.

Syntax	<code>public String setNumber (String value);</code>

Parameters	<i>value</i> — The credit card number.

Discussion

getDescription()

Retrieves the credit card description.

Syntax	<code>public String getDescription();</code>

Parameters	None.
Discussion	

setDescription()
Specifies the credit card description.

Syntax	<code>public void setDescription (String <i>value</i>) ;</code>
Parameters	<i>value</i> — The credit card description.
Discussion	

getExpDate()
Retrieves the credit card expiration date.

Syntax	<code>public String getExpDate() ;</code>
Parameters	None.
Discussion	

setExpDate()
Specifies the credit card expiration date.

Syntax	<code>public void setExpDate (String <i>value</i>) ;</code>
Parameters	<i>value</i> — The credit card expiration date.
Discussion	

getOwner()
Retrieves the credit card owner's name.

Syntax	<code>public String getOwner() ;</code>
Parameters	None.
Discussion	

setOwner()

Specifies the credit card owner's name.

Syntax	<code>public void setOwner (String value) ;</code>

Parameters	<i>value</i> — The credit card owner's name.

Discussion

getBillingName()

Retrieves the credit card billing name.

Syntax	<code>public String getBillingName() ;</code>

Parameters	None.

Discussion

setBillingName()

Specifies the credit card billing name.

Syntax	<code>public void setBillingName (String value) ;</code>

Parameters	<i>value</i> — The billing name for the credit card.

Discussion

getAddress()

Retrieves the credit card address.

Syntax	<code>public String getAddress() ;</code>

Parameters	None.

Discussion

setAddress()

Specifies the credit card address.

Syntax	<code>public void setBillingName (String value) ;</code>

Parameters *value* — The credit card address.

Discussion

getCity()
Retrieves the credit card address city name.

Syntax `public String getCity();`

Parameters None.

Discussion

setCity()
Specifies the credit card address city name.

Syntax `public void setCity (String value);`

Parameters *value* — The credit card address city name.

Discussion

getState()
Retrieves the credit card address state name.

Syntax `public void getState();`

Parameters None.

Discussion

setState()
Specifies the credit card address state name.

Syntax	<code>public void setState (String value);</code>

Parameters	<i>value</i> — The credit card address state name.

Discussion

getCountry()

Retrieves the credit card address country name.

Syntax	<code>public String setCountry();</code>

Parameters	None.

Discussion

setCountry()

Specifies the credit card address country name.

Syntax	<code>public void setCountry (String value);</code>

Parameters	<i>value</i> — The credit card address country name.

Discussion

getPostalCode()

Retrieves the credit card address postal code.

Syntax	<code>public String getPostalCode();</code>

Parameters	None.

Discussion

setPostalCode()

Specifies the credit card address postal code.

Syntax	<code>public void setPostalCode (String value);</code>

Parameters	<i>value</i> — The credit card address postal code.

Discussion

getCardStatus()
Retrieves the credit card status.

Syntax	<code>public String getCardStatus();</code>
Parameters	None.
Discussion	

setCardStatus()
Retrieves the credit card status.

Syntax	<code>public void setCardStatus (String value);</code>
Parameters	<i>value</i> — The credit card status.
Discussion	

getPin()
Retrieves the credit card pin number.

Syntax	<code>public String getPin();</code>
Parameters	None.
Discussion	

setPin()
Specifies the credit card pin number.

Syntax	<code>public void setPin (String value);</code>
Parameters	<i>value</i> — The credit card pin number.
Discussion	

getExtDate1()

Retrieves the first extension date.

Syntax	<code>public Date getExtDate1();</code>

Parameters	None.

Discussion	

setExtDate1()

Specifies the credit card first extension date.

Syntax	<code>public void setExtDate1 (Date <i>date</i>);</code>

Parameters	<i>date</i> — The first extension date.

Discussion	

getExtDate2()

Retrieves the second extension date.

Syntax	<code>public Date getExtDate2();</code>

Parameters	None.

Discussion	

setExtDate2()

Specifies the second extension date.

Syntax	<code>public void setExtDate1 (Date <i>date</i>);</code>

Parameters	<i>date</i> — The second extension date.

Discussion	

getExtDate3()

Retrieves the third extension date.

Syntax	<code>public Date getExtDate3();</code>

Parameters	None.
Discussion	

setExtDate1()
Specifies the third extension date.

Syntax	<code>public void setExtDate3 (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — The third extension date.
Discussion	

getExtDate4()
Retrieves the fourth extension date.

Syntax	<code>public Date getExtDate4 () ;</code>
Parameters	None.
Discussion	

setExtDate4()
Specifies the fourth extension date.

Syntax	<code>public void setExtDate4 (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — The fourth extension date.
Discussion	

getExtNum1()
Retrieves the first extension number.

Syntax	<code>public double getExtNum1 () ;</code>
Parameters	None.
Discussion	

setExtNum1()

Specifies the first extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — The first extension number.
Discussion	

getExtNum2()

Retrieves the second extension number.

Syntax	<code>public double getExtNum2();</code>
Parameters	None.
Discussion	

setExtNum2()

Specifies the second extension number.

Syntax	<code>public void setExtNum2 (double <i>num</i>);</code>
Parameters	<i>num</i> — The second extension number.
Discussion	

getExtNum3()

Retrieves the third extension number.

Syntax	<code>public double getExtNum3();</code>
Parameters	None.
Discussion	

setExtNum3()

Specifies the third extension number.

Syntax	<code>public void setExtNum3 (double <i>num</i>);</code>

Parameters	<i>num</i> — The third extension number.
Discussion	

getExtNum4()
Retrieves the fourth extension number.

Syntax	<code>public double getExtNum4();</code>
Parameters	None.
Discussion	

setExtNum4()
Specifies the fourth extension number.

Syntax	<code>public void setExtNum4 (double <i>num</i>);</code>
Parameters	<i>num</i> — The fourth extension number.
Discussion	

BXContainer Class Reference

The BXContainer class is the base class for defining methods that manipulate objects representing container records.

Whenever possible, use the interface class IBXContainer (see “IBXContainer Class Reference”) instead of BXContainer.

IBXContainer Class Reference

The IBXContainer class is an interface for a document container. A bill container has its own summary and can also contain many billing documents. A bill container record is associated with a summary record and zero or more document records.

This class defines a method that allows access to related document records. You can also call the `BXBillingMgr.createContainer()` method to create a container object.

Class Definition

Table 2-12 describes the IBXContainer interface definition.

Table 2-12 IBXContainer Class Definition

Package	<code>com.ipplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public interface IBXContainer extends IBXBillingEntity {...}</code>

Methods

This section describes the IBXContainer class methods. Table 2-13 lists these methods.

Table 2-13 IBXContainer Class Methods

Method	Description
Manipulating container-related records	
<code>ejbFindAllDocuments()</code>	Retrieves the documents associated with this container.
<code>getSummary()</code>	Retrieves the summary information associated with this container.
Getting and setting container information	
<code>getAccountID()</code>	Retrieves the container's account ID.
<code>setAccountID()</code>	Specifies the container's account ID.
<code>getFileID()</code>	Retrieves the file ID.
<code>setFileID()</code>	Specifies the file ID.
<code>getDescription()</code>	Retrieves the container's description.
<code>setDescription()</code>	Specifies the container's description.
<code>getHeader()</code>	Retrieves the container's header.
<code>setHeader()</code>	Specifies the container's header.
<code>getFooter()</code>	Retrieves the container's footer.
<code>setFooter()</code>	Specifies the container's footer.
<code>getExtension()</code>	Retrieves the container's extension.

Table 2-13 IBXContainer Class Methods

Method	Description
setExtension()	Specifies the container's extension.
getExtDate1()	Retrieves the first extension date
setExtDate1()	Specifies the first extension date.
getExtDate2()	Retrieves the second extension date
setExtDate2()	Specifies the second extension date.
getExtDate3()	Retrieves the third extension date
setExtDate3()	Specifies the third extension date.
getExtDate4()	Retrieves the fourth extension date
setExtDate4()	Specifies the fourth extension date.
getExtNum1()	Retrieves the first extension number.
setExtNum1()	Specifies the first extension number.
getExtNum2()	Retrieves the second extension number.
setExtNum2()	Specifies the second extension number.
getExtNum3()	Retrieves the third extension number.
setExtNum3()	Specifies the third extension number.
getExtNum4()	Retrieves the fourth extension number.
setExtNum4()	Specifies the fourth extension number.

ejbFindAllDocuments()

Retrieves the documents associated with this container.

Syntax	<code>public Vector ejbFindAllDocuments(BXFilter <i>filter</i>, BXOrder <i>order</i>) throws Exception;</code>
Parameters	<i>filter</i> — A BXFilter object that specifies the Where clause of an SQL Select statement. If null, all records will be retrieved. <i>order</i> — A BXOrder object that specifies the Order By clause of an SQL Select statement. If null, the retrieved records will be sorted in ascending order by their primary key.

Discussion	Retrieves all document records for the container that match the clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, the <code>ejbFindAllDocuments()</code> method creates an object, specifies the fields in the object from the record, and adds the object to the return vector. The container object's database context must previously have been set by calling each object's <code>createBillingContext()</code> method. The account ID must be set in the container object. Returns a vector that contains <code>BXDocument</code> objects representing the selected records

getAccountID()

Retrieves the container's account ID.

Syntax	<code>publicString getAccountID();</code>
Parameters	None.
Discussion	Returns a string that contains the account ID.

getDescription()

Retrieves the container's description.

Syntax	<code>public String getDescription();</code>
Parameters	None.
Discussion	Returns a string that contains the description.

getExtension()

Retrieves the container's extension.

Syntax	<code>public String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the extension.

getFileID()

Retrieves the file ID.

Syntax	<code>public String getFileID();</code>
Parameters	None.
Discussion	Returns a string that contains the file ID.

getFooter()

Specifies the container's footer.

Syntax	<code>public String getFooter();</code>
Parameters	None.
Discussion	Returns a string that contains the footer.

getHeader()

Specifies the container's header.

Syntax	<code>public String getHeader();</code>
Parameters	None.
Discussion	Returns a string that contains the header.

getSummary()

Retrieves the summary information associated with this document container.

Syntax	<code>public IBXSummary getSummary();</code>
Parameters	None.
Discussion	Retrieves an IBXSummary object that contains the summary information about the document container.

setAccountID()

Specifies the container's account ID.

Syntax	<code>public void setAccountID (String <i>accountID</i>) ;</code>
Parameters	<i>accountID</i> — A string the specifies the ID
Discussion	Only changes the value of the account ID field in the object; it does not modify the value in the database. Returns an IBXSummary object that contains the summary.

setDescription()

Specifies the container's description.

Syntax	<code>public void setDescription(String <i>description</i>) ;</code>
Parameters	<i>description</i> — A string that specifies the description
Discussion	Only changes the value of the description field in the object; it does not modify the value in the database.

setExtension()

Specifies the container's extension.

Syntax	<code>public void setExtension (String <i>extension</i>) ;</code>
Parameters	<i>extension</i> — A string that specifies the extension.
Discussion	Only changes the value of the description field in the object; it does not modify the value in the database.

setFileID()

Specifies the file ID.

Syntax	<code>public void setFileID (String <i>fileID</i>) ;</code>
Parameters	<i>fileID</i> — A string that specifies the file ID
Discussion	Only changes the value of the extension field in the object; it does not modify the value in the database.

setFooter()

Specifies the container’s footer.

Syntax	<code>public void setFooter (String <i>footer</i>) ;</code>
Parameters	<i>footer</i> — A string that contains the footer
Discussion	Only changes the value of the footer field in the object; it does not modify the value in the database.

setHeader()

Specifies the container’s header.

Syntax	<code>public void setHeader (String <i>header</i>) ;</code>
Parameters	<i>header</i> — A string that contains the header
Discussion	Only changes the value of the header field in the object; it does not modify the value in the database.

getExtDate1()

Retrieves the first extension date.

Syntax	<code>public Date getExtDate1() ;</code>
Parameters	None.
Discussion	

setExtDate1()

Specifies the first extension date.

Syntax	<code>public void setExtDate1(Date <i>date</i>) ;</code>
Parameters	<i>date</i> — A date that indicates the first extension date.
Discussion	

getExtDate2()

Retrieves the second extension date.

Syntax	<code>public Date getExtDate2();</code>

Parameters	None.

Discussion	

setExtDate2()

Specifies the second extension date.

Syntax	<code>public void setExtDate2 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the second extension date.

Discussion	

getExtDate3()

Retrieves the third extension date.

Syntax	<code>public Date getExtDate3();</code>

Parameters	None.

Discussion	

setExtDate3()

Specifies the third extension date.

Syntax	<code>public void setExtDate3 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the third extension date.

Discussion	

getExtDate4()

Retrieves the fourth extension date.

Syntax	<code>public Date getExtDate4();</code>

Parameters	None.
Discussion	

setExtDate4()
Specifies the fourth extension date.

Syntax	<code>public void setExtDate4 (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — A date that indicates the fourth extension date.
Discussion	

getExtNum1()
Retrieves the first extension number.

Syntax	<code>public double getExtNum1 () ;</code>
Parameters	None.
Discussion	

setExtNum1()
Specifies the first extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that specifies the extension number.
Discussion	

getExtNum2()
Retrieves the second extension number.

Syntax	<code>public double getExtNum2 () ;</code>
Parameters	None.
Discussion	

setExtNum2()

Specifies the second extension number.

Syntax	<code>public void setExtNum2 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that specifies the extension number.
Discussion	

getExtNum3()

Retrieves the third extension number.

Syntax	<code>public double getExtNum3();</code>
Parameters	None.
Discussion	

setExtNum3()

Specifies the third extension number.

Syntax	<code>public void setExtNum3 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that specifies the extension number.
Discussion	

getExtNum4()

Retrieves the fourth extension number.

Syntax	<code>public double getExtNum4();</code>
Parameters	None.
Discussion	

setExtNum4()

Specifies the fourth extension number.

Syntax	<code>public void setExtNum4 (double <i>num</i>);</code>

Parameters	<i>num</i> — A double that specifies the extension number.
Discussion	

BXDetail Class Reference

The BXDetail base class manipulates detail line items.

Whenever possible, use the interface class IBXDetail (see “IBXDetail Class Reference”) instead of BXDetail.

IBXDetail Class Reference

The IBXDetail class is the interface for detail line items.

Class Definition

Table 2-14 describes the class definition for the IBXDetail interface. Table 2-14 lists these methods.

Table 2-14 IBXDetail Class Definition	
Package	com.iplanet.ecomm.billxb2b.ejb.billing;
Syntax	public interface IBXDetail extends IBXBillingEntity {...}

Methods

This section describes the IBXDetail class methods. Table 2-15 lists these methods.

Table 2-15 IBXDetail Class Methods (1 of 2)	
Method	Description
Getting and setting detailed document information	
getDocID()	Retrieves the detail line’s document ID.
setDocID()	Specifies the detail line’s document ID.
getFileID()	Retrieves the file ID.
setFileID()	Specifies the file ID.

Table 2-15 IBXDetail Class Methods (2 of 2)

Method	Description
<code>getDescription()</code>	Retrieves the detail line's description.
<code>setDescription()</code>	Specifies the detail line's description.
<code>getLineItemType()</code>	Retrieves the detail line's type.
<code>setLineItemType()</code>	Specifies the detail line's type.
<code>getLineNumber()</code>	Retrieves the line number.
<code>setLineNumber()</code>	Specifies the line number.

getDocID

Retrieves the detail line's document ID.

Syntax	<code>public String getDocID();</code>
Parameters	None.
Discussion	Returns a string that contains the document ID.

setDocID

Specifies the detail line's document ID.

Syntax	<code>public void setDocID (String <i>docID</i>);</code>
Parameters	<i>docID</i> — A string that specifies the document ID.
Discussion	Only changes the value of the object field; it does not modify the database value.

getFileID

Retrieves the file ID.

Syntax	<code>public String getFileID();</code>
Parameters	None.
Discussion	Returns a string that contains the file ID.

setFileID

Specifies the file ID.

Syntax	<code>public void setFileID (String <i>fileID</i>) ;</code>
Parameters	<i>fileID</i> — A string that contains the file ID.
Discussion	Only changes the value of the object field; it does not modify the database value.

getLineItemType()

Retrieves the detail's line item type.

Syntax	<code>public String getLineItemType() ;</code>
Parameters	None.
Discussion	Returns a string that contains the line item type.

setLineItemType()

Specifies the detail's line item type.

Syntax	<code>public void setLineItemType(String <i>lineItemType</i>) ;</code>
Parameters	<i>lineItemType</i> — The line item type.
Discussion	Only changes the value of the object field; it does not modify the database value.

getLineNumber()

Retrieves the line number.

Syntax	<code>public int getLineNumber() ;</code>
Parameters	None.
Discussion	Returns a string that contains the line number.

setLineNumber()

Specifies the line number.

Syntax	<code>public void setLineNumber(int <i>Num</i>);</code>
Parameters	<i>Num</i> — A string that contains the line number.
Discussion	Only changes the value of the object field; it does not modify the database value.

getDescription()

Retrieves the detail line's description.

Syntax	<code>public String getDescription();</code>
Parameters	None.
Discussion	Only changes the value of the object field; it does not modify the database value.

setDescription()

Specifies the detail line's description.

Syntax	<code>public void setDescription (String <i>description</i>);</code>
Parameters	<i>description</i> — A string that specifies the description.
Discussion	Only changes the value of the object field; it does not modify the database value.

BXDocument Class Reference

The BXDocument class is the base class for defining methods that manipulate objects representing billing records.

Whenever possible, use the interface class IBXDocument (see “IBXDocument Class Reference”) instead of BXDocument.

IBXDocument Class Reference

The IBXDocument interface class defines methods that manipulate objects representing document bills. You call the `BXBillingMgr.createDocument()` method to create a document object.

Before you can call the `IBXEntity.ejbCreate()` method (see “IBXBillingEntity Class Reference,” on page 164) to create a new record associated with the object, you must call the `setAccountID()` method to set the account ID.

Class Definition

Table 2-16 describes the IBXDocument interface class definition.

Table 2-16 BXDocument Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public interface IBXDocument extends IBXBillingEntity {...}</code>

Methods

This section describes the IBXDocument methods. Table 2-17 lists these methods.

Table 2-17 IBXDocument Class Methods (1 of 3)

Methods	Description
Manipulating billing records	
<code>ejbFindAllDetailsByType()</code>	Retrieves detail records of the specified type associated with this bill.
<code>ejbFindAllPaymentHistories()</code>	Retrieves payment history records associated with this bill.
<code>getSummary()</code>	Retrieves the summary associated with this bill.
Getting and setting billing information	
<code>getContainerID()</code>	Retrieves the bill’s container ID.
<code>setContainerID()</code>	Specifies the bill’s container ID.
<code>getAccountID()</code>	Retrieves the bill’s account ID.
<code>setAccountID()</code>	Specifies the bill’s account ID.
<code>getFileID()</code>	Retrieves the file ID.
<code>setFileID()</code>	Specifies the file ID.

Table 2-17 IBXDocument Class Methods (2 of 3)

Methods	Description
<code>getDescription()</code>	Retrieves the bill's description.
<code>setDescription()</code>	Specifies the bill's description.
<code>getDocType()</code>	Retrieves the bill type.
<code>setDocType()</code>	Specifies the bill type.
<code>getBillType()</code>	Retrieves the bill's bill type.
<code>setBillType()</code>	Specifies the bill's bill type.
<code>getCreateDate()</code>	Retrieves the bill's creation date.
<code>getModifyDate()</code>	Retrieves the bill's modification date.
<code>getFirstViewDate()</code>	Retrieves the bill's first view date.
<code>setFirstViewDate()</code>	Specifies the bill's first view date.
<code>getLastViewDate()</code>	Retrieves the bill's last view date.
<code>setLastViewDate()</code>	Specifies the bill's last view date.
<code>getViewCount()</code>	Retrieves the bill's view count.
<code>setViewCount()</code>	Specifies the bill's view count.
<code>getDocStatus()</code>	Retrieves the bill's status.
<code>setDocStatus()</code>	Specifies the bill's status.
<code>getCopyType()</code>	Retrieves the bill's copy type.
<code>setCopyType()</code>	Specifies the bill's copy type.
<code>getExtension()</code>	Retrieves the bill's extension.
<code>setExtension()</code>	Specifies the bill's extension.
<code>getBillDate()</code>	Retrieves the bill's bill date.
<code>setBillDate()</code>	Specifies the bill's bill date ID.
<code>getCanView()</code>	Determines if the bill is viewable
<code>setCanView()</code>	Specifies if the bill is viewable
<code>getInvoiceNum()</code>	Retrieves the invoice number.
<code>setInvoiceNum()</code>	Specifies the invoice number.
<code>incViewCount()</code>	Updates the view count and the last view date.
<code>getPMPrStatus()</code>	Retrieves the price status.
<code>setPMPrStatus()</code>	Specifies the price status.

Table 2-17 IBXDocument Class Methods (3 of 3)

Methods	Description
getPoNum()	Retrieves the purchase order number.
setPoNum()	Specifies the purchase order number.
getDeptNum()	Retrieves the department order number.
setDeptNum()	Specifies the department order number.
getExtDate1()	Retrieves the first extension date
setExtDate1()	Specifies the first extension date.
getExtDate2()	Retrieves the second extension date
setExtDate2()	Specifies the second extension date.
getExtDate3()	Retrieves the third extension date
setExtDate3()	Specifies the third extension date.
getExtDate4()	Retrieves the fourth extension date
setExtDate4()	Specifies the fourth extension date.
getExtNum1()	Retrieves the first extension number.
setExtNum1()	Specifies the first extension number.
getExtNum2()	Retrieves the second extension number.
setExtNum2()	Specifies the second extension number.
getExtNum3()	Retrieves the third extension number.
setExtNum3()	Specifies the third extension number.
getExtNum4()	Retrieves the fourth extension number.
setExtNum4()	Specifies the fourth extension number.

ejbFindAllDetailsByType()

Retrieves detail records of the specified type associated with this bill.

Syntax	<code>public Vector ejbFindAllDetailsByType(String type, BXFilter filter, BXOrder order) throws Exception;</code>

Parameters	<i>type</i> — A string that specifies a detail record’s type. <i>filter</i> — A BXFilter object that specifies the Where clause of an SQL Select statement. If null, all records will be retrieved. <i>order</i> — A BXOrder object that specifies the Order By clause of an SQL Select statement. If null, the retrieved records will be sorted in ascending order by their primary key.
Discussion	Retrieves all detail records associated with the bill that match type and clause in the <i>filter</i> parameter, ordered by the clause specified in the order parameter. For each record, creates an object, specifies the fields in the object from the record, and adds the object to the return vector. You can specify the following types: TelCoCall — Retrieves call detail records for IBXDetailCall TelCoMisc — Retrieves detail records for IBXDetailMisc TelCoRating — Retrieves rating detail records for IBXDetailRating The bill object’s database context must previously have been set by calling each object’s <code>createBillingContext()</code> method. The account ID must be set in the bill object. Returns a vector that contains objects representing the selected records.

ejbFindAllPaymentHistories()

Retrieves payment history records associated with this bill, with the most recent ones first.

Syntax	<pre>public Vector ejbFindAllPaymentHistories (BXFilter <i>filter</i>, BXOrder <i>order</i>) throws Exception;</pre>
Parameters	<i>filter</i> — If null, all records will be retrieved. <i>order</i> — If null, the retrieved records will be sorted in ascending order by their primary key.
Discussion	Retrieves all payment history records associated with the bill that match the specified clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, creates an object, specifies the fields in the object from the record, and adds the object to the return vector. The bill object’s database context must previously have been set by calling each object’s <code>createBillingContext()</code> method. The account ID must be set in the bill object. Returns a vector that contains IBXPayHistory objects representing the selected records.

getAccountID()

Retrieves the bill’s account ID.

Syntax	<code>public String getAccountID();</code>
Parameters	None.
Discussion	Returns a string that contains the account ID.

getBillDate()

Retrieves the bill’s bill date.

Syntax	<code>public Date getBillDate ();</code>
Parameters	None.
Returns	Returns a Date object that contains the date.

getBillType()

Retrieves the bill’s bill type.

Syntax	<code>public String getBillType ();</code>
Parameters	None.
Discussion	Returns a string that contains the type.

getCanView()

Determines if the bill is viewable.

Syntax	<code>public String getCanView();</code>
Parameters	None.
Discussion	Returns a string that determines if the bill is viewable.

getContainerID()

Retrieves the bill's container ID.

Syntax	<code>public String getContainerID();</code>
Parameters	None.
Discussion	Returns a string that contains the container ID.

getCopyType()

Retrieves the bill's copy type.

Syntax	<code>public String getCopyType();</code>
Parameters	None.
Discussion	Returns a string that contains the type.

getCreateDate()

Retrieves the bill's creation date.

Syntax	<code>public Date getCreateDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the date.

getDescription()

Retrieves the bill's description.

Syntax	<code>public String getDescription();</code>
Parameters	None.
Discussion	Returns a string that contains the description.

getDocStatus()

Retrieves the bill's status.

Syntax	<code>public String getDocStatus();</code>

Parameters	None.
Discussion	Returns a string that contains the status.

getDocType()
Retrieves the bill type.

Syntax	<code>public String getDocType();</code>
Parameters	None.
Discussion	Returns a string that contains the bill type.

getExtension()
Retrieves the bill’s extension.

Syntax	<code>public String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the extension ID.

getFileID()
Retrieves the file ID.

Syntax	<code>public String getFileID();</code>
Parameters	None.
Discussion	Returns a string that contains the ID.

getFirstViewDate()
Retrieves the bill’s first view date.

Syntax	<code>public Date getFirstViewDate();</code>
Parameters	None.
Discussion	Returns a date object that contains the date.

getLastViewDate()

Retrieves the bill's last view date.

Syntax	<code>public Date getLastViewDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the date.

getModifyDate()

Retrieves the bill's modification date.

Syntax	<code>public Date getModifyDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the modification date.

getSummary()

Retrieves the summary associated with this bill.

Syntax	<code>public IBXSummary getSummary();</code>
Parameters	None.
Discussion	Returns an IBXSummary object that contains the summary.

getViewCount()

Retrieves the bill's view count.

Syntax	<code>public int getViewCount();</code>
Parameters	None.
Discussion	Returns an integer that contains the count.

setAccountID()

Specifies the bill's account ID.

Syntax	<code>public void setAccountID (String accountID);</code>

Parameters	<i>accountID</i> — A string that specifies the ID
Discussion	Only changes the value of the object field; it does not modify the database value.

setBillDate()
Specifies the bill’s bill date.

Syntax	<code>public void setBillDate (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — A Date object that specifies the date
Discussion	Only changes the value of the bill date field in the object; it does not modify the value in the database.

setBillType()
Specifies the bill type.

Syntax	<code>public void setBillType (String <i>billType</i>) ;</code>
Parameters	<i>billType</i> — A string that specifies the bill type.
Discussion	Only changes the value of the object field; it does not modify the database value.

setCanView()
Specifies if the bill is viewable.

Syntax	<code>public void setCanView (String <i>canView</i>) ;</code>
Parameters	<i>canView</i> — A string that specifies if the bill is viewable.
Discussion	Only changes the value of the object field; it does not modify the database value.

setDocStatus()
Specifies the bill’s status

Syntax	<code>public void setDocStatus(String <i>status</i>) ;</code>

Parameters	<i>status</i> — A string that specifies the status
Discussion	Only changes the value of the status field in the object; it does not modify the value in the database.

setDocType()

Specifies the bill type.

Syntax	<code>public void setDocType(String <i>docType</i>);</code>
Parameters	<i>docType</i> — A string that specifies the doc type
Discussion	Only changes the value of the doc type field in the object; it does not modify the value in the database.

setExtension()

Specifies the bill's extension.

Syntax	<code>public void setExtension(String <i>extension</i>);</code>
Parameters	<i>extension</i> — A string that specifies the extension
Discussion	Only changes the value of the extension field in the object; it does not modify the value in the database.

setFileID()

Specifies the file ID.

Syntax	<code>public void setFileID(String <i>fileID</i>);</code>
Parameters	<i>fileID</i> — A string that specifies the file ID
Discussion	Only changes the value of the file ID field in the object; it does not modify the value in the database.

setFirstViewDate()

Specifies the bill’s first view date.

Syntax	<code>public void setFirstViewDate(Date <i>date</i>);</code>
Parameters	<i>date</i> — A string that specifies the date
Discussion	Only changes the value of the object field; it does not modify the database value.

setLastViewDate()

Specifies the bill’s last view date.

Syntax	<code>public void setLastViewDate(Date <i>date</i>);</code>
Parameters	<i>date</i> — A string that specifies the date
Discussion	Only changes the value of the object field; it does not modify the database value.

setViewCount()

Specifies the bill’s view count.

Syntax	<code>public void setViewCount (int <i>viewCount</i>);</code>
Parameters	<i>viewCount</i> — A string that specifies the view count
Discussion	Only changes the value of the view count field in the object; it does not modify the value in the database.

getExtension()

Retrieves the detail line’s extension.

Syntax	<code>public String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the Extension.

getFileID()

Retrieves the file ID.

Syntax	<code>public String getFileID();</code>
Parameters	None.
Discussion	Returns a string that contains the ID.

setDescription()

Specifies the detail line's description.

Syntax	<code>public void setDescription (String <i>description</i>);</code>
Parameters	<i>description</i> — A string that specifies the description.
Discussion	Only changes the value of the object field; it does not modify the database value.

setFileID()

Specifies the file ID.

Syntax	<code>public void setFileID (String <i>fileID</i>);</code>
Parameters	<i>fileID</i> — A string that specifies the file ID
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

getInvoiceNum()

Retrieves the invoice number.

Syntax	<code>public String getInvoiceNum();</code>
Parameters	None.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

setInvoiceNum()

Specifies the invoice number.

Syntax	<code>public void setInvoiceNum (String <i>invoiceNum</i>) ;</code>
Parameters	<i>invoiceNum</i> — A string that specifies the invoice number.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

incViewCount()

Updates the view count and the last view date.

Syntax	<code>public void incViewCount() throws Exception;</code>
Parameters	None.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

getPMPrStatus()

Retrieves the price status.

Syntax	<code>public String getPMPrStatus();</code>
Parameters	None.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

setPMPrStatus()

Specifies the price status.

Syntax	<code>public void setPMPrStatus (String <i>pmprStatus</i>);</code>
Parameters	<i>pmprStatus</i> — A string that specifies the price status.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

getPoNum()

Retrieves the purchase order number.

Syntax	<code>public String getPMPrStatus();</code>
Parameters	None.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

setPoNum()

Specifies the purchase order number.

Syntax	<code>public void setPMPrStatus (String <i>poNum</i>);</code>
Parameters	<i>poNum</i> — A string that specifies the purchase order number.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

getDeptNum()

Retrieves the department number.

Syntax	<code>public String getDeptNum();</code>
Parameters	None.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

setDeptNum()

Specifies the department number.

Syntax	<code>public void setDeptNum (String <i>deptNum</i>);</code>
Parameters	<i>deptNum</i> — A string that specifies the department number.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

getExtDate1()

Retrieves the first extension date.

Syntax	<code>public Date getExtDate1();</code>
Parameters	None.
Discussion	

setExtDate1()

Specifies the first extension date.

Syntax	<code>public void setExtDate1 (Date <i>date</i>);</code>
Parameters	<i>date</i> — A date that indicates the first extension date.
Discussion	

getExtDate2()

Retrieves the second extension date.

Syntax	<code>public Date getExtDate2();</code>
Parameters	None.
Discussion	

setExtDate2()

Specifies the second extension date.

Syntax	<code>public void setExtDate2 (Date <i>date</i>);</code>
Parameters	<i>date</i> — A date that indicates the second extension date.
Discussion	

getExtDate3()

Retrieves the third extension date.

Syntax	<code>public Date getExtDate3();</code>

Parameters	None.

Discussion

setExtDate3()

Specifies the third extension date.

Syntax	<code>public void setExtDate3 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the third extension date.

Discussion

getExtDate4()

Retrieves the fourth extension date.

Syntax	<code>public Date getExtDate4();</code>

Parameters	None.

Discussion

setExtDate4()

Specifies the fourth extension date.

Syntax	<code>public void setExtDate4 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the fourth extension date.

Discussion

getExtNum1()

Retrieves the first extension number.

Syntax	<code>public double getExtNum1();</code>

Parameters	None.

Discussion

setExtNum1()

Specifies the first extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the first extension number.
Discussion	

getExtNum2()

Retrieves the second extension number.

Syntax	<code>public double getExtNum2();</code>
Parameters	None.
Discussion	

setExtNum2()

Specifies the second extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the second extension number.
Discussion	

getExtNum3()

Retrieves the third extension number.

Syntax	<code>public double getExtNum3();</code>
Parameters	None.
Discussion	

setExtNum3()

Specifies the third extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>

Parameters	<i>num</i> — A double that indicates the third extension number.
Discussion	

getExtNum4()
Retrieves the fourth extension number.

Syntax	<code>public double getExtNum4 ();</code>
Parameters	None.
Discussion	

setExtNum4()
Specifies the fourth extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the fourth extension number.
Discussion	

BXEventLog Class Reference

The BXEventLog base class defines methods that manipulate objects representing event log entries.

Whenever possible, use the interface class IBXEventLog (see “IBXEventLog Class Reference”) instead of BXEventLog.

IBXEventLog Class Reference

The IBXEventLog class is an interface that defines methods that manipulate objects representing event log entries. You call the `BXBillingMgr.createEventLog()` method to create an event log object.

Before you can call the `IBXEntity.ejbCreate()` (see “IBXBillingEntity Class Reference,” on page 164) method to create a new record associated with the object, you must call the `setAcctID()` method to set the account ID.

Class Definition

Table 2-18 describes the IBXEventLog class definition.

Table 2-18 IBXEventLog Class Definition	
Syntax	com.iplanet.ecomm.billxb2b.ejb.billing;
Package	public interface IBXEventLog extends IBXBillingEntity {...}

Methods

This section describes the IBXEventLog methods. Table 2-19 lists these methods.

Table 2-19 IBXEventLog Methods	
Methods	Description
Getting and setting event log information	
ejbFindEventDetails()	Retrieve event log detail records associated with this event log entry.
getApplName()	Retrieves the application name.
setApplName()	Specifies the application name.
getSubject()	Retrieves the subject.
setSubject()	Specifies the subject.
getAction()	Retrieves the action.
setAction()	Specifies the action.
getOrgID()	Retrieves the organization ID.
setOrgID()	Specifies the organization ID.
getFileID()	Retrieves the File ID.
setFileID()	Specifies the File ID.
getAcctID()	Retrieves the account ID.
setAcctID()	Specifies the account ID.
getCreatedBy()	Retrieves the log entry's creator.
setCreatedBy()	Specifies the log entry's creator.
getMessage()	Retrieves the message.
setMessage()	Specifies the message.

ejbFindEventDetails()

Retrieves event log detail records associated with this event log entry.

Syntax	<code>public Vector ejbFindEventDetails() throws Exception;</code>
Parameters	None.
Discussion	Retrieves all event log detail records in which the detail record's event log ID matches this log entry's account ID. For each record, the <code>ejbFindEventDetails()</code> method creates an <code>BXEventLogDetail</code> object, specifies the name and value fields in the object from the record, and adds the object to the return vector. The account ID must be set in this log entry object and the log entry's object's database context must previously have been set by calling this object's <code>createBillingContext()</code> method. Returns a vector that contains <code>BXEventLogDetail</code> objects

getAcctID()

Retrieves the account ID.

Syntax	<code>public String getAcctID();</code>
Parameters	None.
Discussion	Returns string that specifies the account ID.

getAction()

Retrieves the action.

Syntax	<code>public String getAction();</code>
Parameters	None.
Discussion	Returns string that specifies the action.

getApplName()

Retrieves the application name.

Syntax	<code>public String getApplName();</code>
Parameters	None.

Discussion	Returns a string that specifies the application name.

getCreatedBy()
Retrieves the log entry's creator.

Syntax	<code>public String getCreatedBy();</code>
Parameters	None.
Discussion	Returns a string that specifies the creator.

getFileID()
Retrieves the file ID.

Syntax	<code>public Date getFileID();</code>
Parameters	None.
Discussion	Returns a Date object that specifies the file ID.

getMessage()
Retrieves the message.

Syntax	<code>public String getMessage();</code>
Parameters	None.
Discussion	Returns a string that specifies the message.

getOrgID()
Retrieves the organization ID.

Syntax	<code>public String getOrgID();</code>
Parameters	None.
Discussion	Returns a string that specifies the organization ID.

getSubject()

Retrieves the subject.

Syntax	<code>public String getSubject();</code>
Parameters	None.
Discussion	Returns a string that specifies the subject.

setAcctID()

Specifies the account ID.

Syntax	<code>public void setAcctID(String <i>acctID</i>);</code>
Parameters	<i>acctID</i> — A string that specifies the account ID
Discussion	Only changes the value of the account ID field in the object; it does not modify the value in the database.

setAction()

Specifies the action.

Syntax	<code>public void setAction(String <i>action</i>);</code>
Parameters	<i>action</i> — A string that specifies the action.
Discussion	Only changes the value of the action field in the object; it does not modify the value in the database.

setApplName()

Specifies the application name.

Syntax	<code>public void setApplName(String <i>applName</i>);</code>
Parameters	<i>applName</i> — A string that specifies the application name.
Discussion	Only changes the value of the object field; it does not modify the database value.

setCreatedBy()

Specifies the log entry’s creator.

Syntax	<code>public void setCreatedBy(String <i>createdBy</i>) ;</code>
Parameters	<i>createdBy</i> — A string that specifies the creator.
Discussion	Only changes the value of the created by field in the object; it does not modify the value in the database.

setMessage()

Specifies the message.

Syntax	<code>public void setMessage(String <i>message</i>) ;</code>
Parameters	<i>message</i> — A string that specifies the message.
Discussion	Only changes the value of the message field in the object; it does not modify the value in the database.

setOrgID()

Specifies the organization ID.

Syntax	<code>public void setOrgID(String <i>orgID</i>) ;</code>
Parameters	<i>orgID</i> — A string that specifies the organization ID.
Discussion	Only changes the value of the organization ID field in the object; it does not modify the value in the database.

setFileID()

Specifies the organization ID.

Syntax	<code>public void setFileID(String <i>fileID</i>) ;</code>
Parameters	<i>fileID</i> — A string that specifies the org ID
Discussion	Only changes the value of the file ID field in the object; it does not modify the value in the database.

getSubject()
Specifies the subject.

Syntax	<code>public void setSubject(String subject);</code>
Parameters	<i>subject</i> — A string that specifies the subject
Discussion	Only changes the value of the subject field in the object; it does not modify the value in the database.

BXEventLogDetail Class Reference

The BXEventLogDetail class defines methods that manipulate objects representing event log details.

Class Description

Table 2-20 describes the BXEventLogDetail class description.

Table 2-20 BXEventLogDetail Class Description

Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public class BXEventLogDetail extends IBXBillingEntity {...}</code>

Methods

This section describes the BXEventLogDetail class methods. Table 2-21 lists these methods.

Table 2-21 BXEventLogDetail Class Methods

Method	Description
<code>getEventLogID()</code>	Retrieves the event log ID.
<code>setEventLogID()</code>	Specifies the event log ID.
<code>getName()</code>	Retrieves a name from the event log.
<code>setName()</code>	Specifies a name from the event log.
<code>getValue()</code>	Retrieves a value from the event log.
<code>setValue()</code>	Specifies a value from the event log.

getEventLogID()

Retrieves the event log ID.

Syntax	<code>public String getEventLogID();</code>
Parameters	None.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database. Returns eventLogID_.

setEventLogID()

Specifies the event log ID.

Syntax	<code>public void setEventLogID (String <i>name</i>);</code>
Parameters	<i>name</i> — A string that specifies the name.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

getName()

Retrieves the name.

Syntax	<code>public String getName();</code>
Parameters	None.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database. Returns name_.

setName()

Specifies the name.

Syntax	<code>public void setEventLogID (String <i>name</i>);</code>
Parameters	<i>name</i> — A string that specifies the name.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

getValue()
Retrieves the value.

Syntax	public String getValue();
Parameters	None.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database. Returns value_.

setValue()
Specifies the value.

Syntax	public void setEventLogID (String value);
Parameters	value — A string that specifies the value.
Discussion	Only changes the value of the field in the object; it does not modify the value in the database.

BXExtensionDesc Class Reference

The BXExtensionDesc base class represents the extension description class.

Whenever possible, use the interface class IBXExtensionDesc (see “IBXExtensionDesc Class Reference) instead of IBXExtensionDesc.

IBXExtensionDesc Class Reference

The IBXExtensionDesc interface class represents the extension description table.

Class Description

Table 2-22 describes the IBXExtensionDesc interface description.

Table 2-22 IBXExtensionDesc Class Description

Package	com.iplanet.ecomm.billxb2b.ejb.billing;
Syntax	public interface IBXExtensionDesc extends IBXBillingEntity {...}

Methods

This section describes the IBXExtensionDesc class methods. Table 2-23 lists these methods.

Table 2-23 IBXExtensionDesc Class Methods

Method	Description
getTableName()	Retrieves the extension description table name.
setTableName()	Specifies the extension description table name
getFieldName()	Retrieves a field name in the extension description table.
setFieldName()	Specifies a field name in the extension description table.
getDescription()	Retrieves a description from the extension description table.
setDescription()	Specifies a description in the extension description table.

getTableName()

Retrieves the extension description table name.

Syntax	<code>public String getTableName();</code>
Parameters	None.
Discussion	

setTableName()

Specifies the extension description table name.

Syntax	<code>public void setTableName(String <i>tableName</i>);</code>
Parameters	<i>tableName</i> — A string that represents the extension description table name.
Discussion	

getFieldName()

Retrieves a field name in the extension description table.

Syntax	<code>public String getFieldName();</code>
Parameters	None.

Discussion

setFieldName()

Specifies a field name in the extension description table.

Syntax	<code>public void setFieldName(String <i>fieldName</i>);</code>
Parameters	<i>fieldName</i> — A string that represents a field name in the extension description table.
Discussion	

getDescription()

Retrieves a description from the extension description table.

Syntax	<code>public String getDescription();</code>
Parameters	None.
Discussion	

setDescription()

Specifies a description in the extension description table.

Syntax	<code>public void setDescription(String <i>fieldName</i>);</code>
Parameters	<i>fieldName</i> — A string that represents a description in the extension description table.
Discussion	

BXLinItem Class Reference

The BXLinItem base class represents objects that relate to purchase order line items.

Whenever possible, use the interface class IBXLinItem (see “IBXLinItem Class Reference”) instead of BXLinItem.

IBXLineItem Class Reference

The IBXLineItem interface class represents objects that relate to purchase order line items.

Class Definition

Table 2-24 describes the IBXLineItem interface class definition.

Table 2-24 IBXLineItem Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public interface IBXLineItem extends IBXDetail {...}</code>

Methods

This section describes the IBXLineItem class methods. Table 2-25 lists these methods.

Table 2-25 IBXLineItem Class Methods (1 of 3)

Method	Summary
Attribute access and modification	
<code>getSkuNumber()</code>	Retrieves the SKU number.
<code>setSkuNumber()</code>	Specifies the SKU number.
<code>getQtyOrdered()</code>	Retrieves the quantity that was ordered.
<code>setQtyOrdered()</code>	Specifies the quantity that was ordered.
<code>getQtyShipped()</code>	Retrieves the quantity to be shipped.
<code>setQtyShipped()</code>	Specifies the quantity to be shipped.
<code>getUnitPrice()</code>	Retrieves the unit price.
<code>setUnitPrice()</code>	Specifies the unit price.
<code>getTotalAmount()</code>	Retrieves the total price.
<code>setTotalAmount()</code>	Specifies the total price.
<code>getTaxable()</code>	Retrieves the tax amount.
<code>setTaxable()</code>	Specifies the tax amount.
<code>getDisputeReason()</code>	Retrieves the dispute reason.

Table 2-25 IBXLineItem Class Methods (2 of 3)

Method	Summary
<code>setDisputeReason()</code>	Specifies the dispute reason.
<code>getDisputeNote()</code>	Retrieves a note about the dispute reason.
<code>setDisputeNote()</code>	Specifies a note about the dispute reason.
<code>getPoNum()</code>	Retrieves the purchase order number.
<code>setPoNum()</code>	Specifies the purchase order number.
<code>getDeptNum()</code>	Retrieves the department number.
<code>setDeptNum()</code>	Specifies the department number.
<code>getInvoiceNum()</code>	Retrieves the invoice number.
<code>setInvoiceNum()</code>	Specifies the invoice number.
<code>getCostCode()</code>	Retrieves the cost code.
<code>setCostCode()</code>	Specifies the cost code.
<code>getApprovalStatus()</code>	Retrieves the approval status.
<code>setApprovalStatus()</code>	Specifies the approval status.
<code>getAppStatModifiedBy()</code>	Retrieves the name of the person that modified the approval status.
<code>setAppStatModifiedBy()</code>	Retrieves the name of the person that modified the approval status.
<code>getPaymentStatus()</code>	Retrieves the payment status.
<code>setPaymentStatus()</code>	Specifies the payment status.
<code>getPayStatModifiedBy()</code>	Retrieves the name of the person that modified the payment status.
<code>setPayStatModifiedBy()</code>	Retrieves the name of the person that modified the payment status.
<code>getExtDate1()</code>	Retrieves the first extension date
<code>setExtDate1()</code>	Specifies the first extension date.
<code>getExtDate2()</code>	Retrieves the second extension date
<code>setExtDate2()</code>	Specifies the second extension date.
<code>getExtDate3()</code>	Retrieves the third extension date
<code>setExtDate3()</code>	Specifies the third extension date.
<code>getExtDate4()</code>	Retrieves the fourth extension date
<code>setExtDate4()</code>	Specifies the fourth extension date.
<code>getExtNum1()</code>	Retrieves the first extension number.
<code>setExtNum1()</code>	Specifies the first extension number.

Table 2-25 IBXLineItem Class Methods (3 of 3)

Method	Summary
getExtNum2 ()	Retrieves the second extension number.
setExtNum2 ()	Specifies the second extension number.
getExtNum3 ()	Retrieves the third extension number.
setExtNum3 ()	Specifies the third extension number.
getExtNum4 ()	Retrieves the fourth extension number.
setExtNum4 ()	Specifies the fourth extension number.
getQtyBackOrdered ()	Retrieves the back ordered quantity.
setQtyBackOrdered ()	Specifies the back ordered quantity.

getSkuNumber()

Retrieves the SKU number.

Syntax	<code>public String getSkuNumber();</code>
Parameters	None.
Discussion	

setSkuNumber()

Specifies the SKU number.

Syntax	<code>public void setSkuNumber(String skuNumber);</code>
Parameters	<i>skuNumber</i> — A string that indicates the SKU number.
Discussion	

getQtyOrdered()

Retrieves the quantity ordered.

Syntax	<code>public String getQtyOrdered();</code>
Parameters	None.
Discussion	

setQtyOrdered()

Specifies the quantity ordered.

Syntax	<code>public void setQtyOrdered(double <i>qtyOrdered</i>);</code>
Parameters	<i>qtyOrdered</i> — A double that indicates the quantity ordered.
Discussion	

getQtyShipped()

Retrieves the quantity that was shipped.

Syntax	<code>public double getQtyShipped();</code>
Parameters	None.
Discussion	

setQtyShipped()

Specifies the quantity that was shipped.

Syntax	<code>public void setQtyShipped(double <i>qtyShipped</i>);</code>
Parameters	<i>skuNumber</i> — A double that indicates the quantity shipped.
Discussion	

getUnitPrice()

Retrieves the unit price.

Syntax	<code>public double getUnitPrice();</code>
Parameters	None.
Discussion	

setUnitPrice()

Specifies the unit price.

Syntax	<code>public void setUnitPrice(double <i>unitPrice</i>);</code>

Parameters	<i>unitPrice</i> — A double that indicates the quantity ordered.
Discussion	

getTotalAmount()
Retrieves the total amount of an order.

Syntax	<code>public double getTotalAmount();</code>
Parameters	None.
Discussion	

setTotalAmount()
Specifies the total amount of an order.

Syntax	<code>public void setTotalAmount(double <i>amount</i>);</code>
Parameters	<i>amount</i> — A double that indicates the total amount of an order.
Discussion	

getTaxable()
Retrieves the taxable amount.

Syntax	<code>public double getTaxable();</code>
Parameters	None.
Discussion	

setTaxable()
Specifies the taxable amount.

Syntax	<code>public void setTaxable(String <i>taxable</i>);</code>
Parameters	<i>amount</i> — A string that indicates the taxable amount.
Discussion	

getDisputeReason()

Retrieves the dispute reason.

Syntax	<code>public double getDisputeReason();</code>

Parameters	None.

Discussion	

setDisputeReason()

Specifies the dispute reason.

Syntax	<code>public void getDisputeReason(String <i>reason</i>);</code>

Parameters	<i>amount</i> — A string that indicates the dispute reason.

Discussion	

getDisputeNote()

Retrieves a note about the dispute reason.

Syntax	<code>public double getDisputeNote();</code>

Parameters	None.

Discussion	

setDisputeNote()

Specifies a note about the dispute reason.

Syntax	<code>public void getDisputeNote(String <i>note</i>);</code>

Parameters	<i>note</i> — A string that represents a note about the dispute reason.

Discussion	

getPoNum()

Retrieves a purchase order number.

Syntax	<code>public String getPoNum();</code>

Parameters	None.
Discussion	

setPoNum()
Specifies a purchase order number.

Syntax	<code>public void setPoNum(String <i>poNum</i>);</code>
Parameters	<i>poNum</i> — A string that indicates a a purchase order number.
Discussion	

getDeptNum()
Retrieves a department number.

Syntax	<code>public String getDeptNum();</code>
Parameters	None.
Discussion	

setDeptNum()
Specifies a department number.

Syntax	<code>public void setDeptNum(String <i>deptNum</i>);</code>
Parameters	<i>deptNum</i> — A string that indicates a department number.
Discussion	

getInvoiceNum()
Retrieves an invoice number.

Syntax	<code>public String getInvoiceNum();</code>
Parameters	None.
Discussion	

setInvoiceNum()

Specifies an invoice number.

Syntax	<code>public void setInvoiceNum(String <i>invoiceNum</i>);</code>
Parameters	<i>deptNum</i> — A string that indicates an invoice number.
Discussion	

getCostCode()

Retrieves a cost code.

Syntax	<code>public String getCostCode();</code>
Parameters	None.
Discussion	

setCostCode()

Specifies a cost code.

Syntax	<code>public void setCostCode(String <i>costCode</i>);</code>
Parameters	<i>costCode</i> — A string that indicates a cost code.
Discussion	

getApprovalStatus()

Retrieves the approval status.

Syntax	<code>public String getApprovalStatus();</code>
Parameters	None.
Discussion	

setApprovalStatus()

Specifies the approval status.

Syntax	<code>public void setApprovalStatus(String <i>approvalStatus</i>);</code>

Parameters	<i>approvalStatus</i> — A string that indicates the approval status.
Discussion	

getAppStatModifiedBy()
Retrieves the person’s name that modified the approval status.

Syntax	<code>public String getAppStatModifiedBy();</code>
Parameters	None.
Discussion	

setAppStatModifiedBy()
Specifies the person’s name that modified the approval status.

Syntax	<code>public void setAppStatModifiedBy(String <i>appStatModifiedBy</i>);</code>
Parameters	<i>appStatModifiedBy</i> — A string that indicates who modified the approval status.
Discussion	

getPaymentStatus()
Retrieves the payment status.

Syntax	<code>public String getPaymentStatus();</code>
Parameters	None.
Discussion	

setPaymentStatus()
Specifies the payment status.

Syntax	<code>public void setPaymentStatus(String <i>paymentStatus</i>);</code>
Parameters	<i>paymentStatus</i> — A string that indicates the payment status.
Discussion	

getPayStatModifiedBy()

Retrieves the person's name that modified the payment status.

Syntax	<code>public String getPayStatModifiedBy();</code>
Parameters	None.
Discussion	

setPayStatModifiedBy()

Specifies the person's name that modified the payment status.

Syntax	<code>public void setPayStatModifiedBy(String <i>payStatModifiedBy</i>);</code>
Parameters	<i>payStatModifiedBy</i> — A string that indicates who modified the approval status.
Discussion	

getExtDate1()

Retrieves the first extension date.

Syntax	<code>public Date getExtDate1();</code>
Parameters	None.
Discussion	

setExtDate1()

Specifies the first extension date.

Syntax	<code>public void setExtDate1(Date <i>date</i>);</code>
Parameters	<i>date</i> — A date that indicates the first extension date.
Discussion	

getExtDate2()

Retrieves the second extension date.

Syntax	<code>public Date getExtDate2();</code>

Parameters	None.
Discussion	

setExtDate2()
Specifies the second extension date.

Syntax	<code>public void setExtDate2 (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — A date that indicates the second extension date.
Discussion	

getExtDate3()
Retrieves the third extension date.

Syntax	<code>public Date getExtDate3 () ;</code>
Parameters	None.
Discussion	

setExtDate3()
Specifies the third extension date.

Syntax	<code>public void setExtDate3 (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — A date that indicates the third extension date.
Discussion	

getExtDate4()
Retrieves the fourth extension date.

Syntax	<code>public Date getExtDate4 () ;</code>
Parameters	None.
Discussion	

setExtDate4()

Specifies the fourth extension date.

Syntax	<code>public void setExtDate4 (Date <i>date</i>);</code>
Parameters	<i>date</i> — A date that indicates the fourth extension date.
Discussion	

getExtNum1()

Retrieves the first extension number.

Syntax	<code>public double getExtNum1();</code>
Parameters	None.
Discussion	

setExtNum1()

Specifies the first extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the first extension number.
Discussion	

getExtNum2()

Retrieves the second extension number.

Syntax	<code>public double getExtNum2();</code>
Parameters	None.
Discussion	

setExtNum2()

Specifies the second extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>

Parameters	<i>num</i> — A double that indicates the second extension number.
Discussion	

getExtNum3()
Retrieves the third extension number.

Syntax	<code>public double getExtNum3 ();</code>
Parameters	None.
Discussion	

setExtNum3()
Specifies the third extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the third extension number.
Discussion	

getExtNum4()
Retrieves the fourth extension number.

Syntax	<code>public double getExtNum4 ();</code>
Parameters	None.
Discussion	

setExtNum4()
Specifies the fourth extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the fourth extension number.
Discussion	

getQtyBackOrdered()

Retrieves the back ordered quantity.

Syntax	<code>public double getQtyBackOrdered();</code>

Parameters	None.

Discussion	

setQtyBackOrdered()

Specifies the back ordered quantity.

Syntax	<code>public void setQtyBackOrdered (double <i>qtyBackOrdered</i>);</code>

Parameters	<i>qtyBackOrdered</i> — A double that represents the quantity that was back ordered.

Discussion	

BXMessage Class Reference

The BXMessage base class defines methods that manipulate objects representing messages.

Whenever possible, use the interface class IBXMessage (see “IBXMessage Class Reference”) instead of IBXMessage.

IBXMessage Class Reference

The IBXMessage interface class defines methods that manipulate objects representing messages. You call the `BXBillingMgr.createMessage()` method to create a message object.

Before you can call the `IBXEntity.ejbCreate()` (see “IBXBillingEntity Class Reference,” on page 164) method to create a new record associated with the object, you must call the `setAcctID()` method to set the account ID.

Class Definition

Table 2-26 describes the IBXMessage class definition.

Table 2-26 IBXMessage Class Definition

Package	com.iplanet.ecomm.billxb2b.ejb.billing;
Syntax	public interface IBXMessage extends IBXBillingEntity {...}

Methods

This section describes the IBXMessage class methods. Table 2-27 lists these methods.

Table 2-27 IBXMessage Class Methods (1 of 2)

Method	Description
Getting and setting message information	
getBillerID()	Retrieves the biller ID.
setBillerID()	Specifies the biller ID.
getDocID()	Retrieves the document ID.
setDocID()	Specifies the document ID.
getAcctID()	Retrieves the account ID.
setAcctID()	Specifies the account ID.
getMsgTo()	Retrieves the message receiver.
setMsgTo()	Specifies the message receiver.
getMsgFrom()	Retrieves the sender.
setMsgFrom()	Specifies the sender.
getSubject()	Retrieves the subject.
setSubject()	Specifies the subject.
getType()	Retrieves the message type.
setType()	Specifies the message type.
getMessage()	Retrieves the message contents.
setMessage()	Specifies the message contents.
getSenderType()	Retrieves the sender type.
setSenderType()	Specifies the sender type.

Table 2-27 IBXMessage Class Methods (2 of 2)

Method	Description
getCaseNum ()	Retrieves the case number.
setCaseNum ()	Specifies the case number.
getCsrName ()	Retrieves the CSR name.
setCsrName ()	Specifies the CSR name.
getMsgStatus ()	Retrieves the status.
setMsgStatus ()	Specifies the status.
getExtension ()	Retrieves the message's extension.
setExtension ()	Specifies the messages extension.
getExtDate1	Retrieves the first extension date
setExtDate1	Specifies the first extension date.
getExtDate2 ()	Retrieves the second extension date
setExtDate2 ()	Specifies the second extension date.
getExtDate3 ()	Retrieves the third extension date
setExtDate3 ()	Specifies the third extension date.
getExtDate4 ()	Retrieves the fourth extension date
setExtDate4 ()	Specifies the fourth extension date.
getExtNum1 ()	Retrieves the first extension number.
setExtNum1 ()	Specifies the first extension number.
getExtNum2 ()	Retrieves the second extension number.
setExtNum2 ()	Specifies the second extension number.
getExtNum3 ()	Retrieves the third extension number.
setExtNum3 ()	Specifies the third extension number.
getExtNum4 ()	Retrieves the fourth extension number.
setExtNum4 ()	Specifies the fourth extension number.

getAcctID()

Retrieves the account ID.

Syntax	<code>public String getAcctID();</code>

Parameters	None.
Discussion	Returns a string that specifies the account ID.

getBillerID()
Retrieves the biller ID.

Syntax	<code>public String getBillerID();</code>
Parameters	None.
Discussion	Returns a string that contains the Biller ID.

getCaseNum()
Retrieves the case number.

Syntax	<code>public String getCaseNum();</code>
Parameters	None.
Discussion	Returns a string that contains the case number.

getCsrName()
Retrieves the message’s creation date.

Syntax	<code>public Date getCsrName ();</code>
Parameters	None.
Discussion	Returns a Date object that contains the CRS name.

getDocID()
Retrieves the document ID.

Syntax	<code>public String getDocID();</code>
Parameters	None.
Discussion	Returns a string that contains the doc ID.

getExtension()

Retrieves the message's extension.

Syntax	<code>public String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the extension.

getMessage()

Retrieves the message contents.

Syntax	<code>public String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the extension.

getMsgFrom()

Retrieves the sender.

Syntax	<code>public String getMsgFrom();</code>
Parameters	None.
Discussion	Returns a string that contains the message sender.

getMsgStatus()

Retrieves the status.

Syntax	<code>public String getMsgStatus();</code>
Parameters	None.
None.	Returns a string that specifies the status.

getMsgTo()

Retrieves the receiver.

Syntax	<code>public String getMsgTo();</code>

Parameters	None.
Discussion	Returns a string that specifies the message receiver.

getSenderType()
Retrieves the sender type.

Syntax	<code>public String getSenderType();</code>
Parameters	None.
Discussion	Returns a string that contains the sender type.

getSubject()
Retrieves the subject.

Syntax	<code>public String getSubject();</code>
Parameters	None.
Discussion	Returns a string that contains the message subject.

getType()
Retrieves the message type.

Syntax	<code>public String getType();</code>
Parameters	None.
Discussion	Returns a string that contains the message type.

setAcctID()
Specifies the account ID.

Syntax	<code>public void setAcctID (String acctID);</code>
Parameters	<i>acctID</i> — A string that specifies the account ID.
Discussion	Only changes the value of the account ID field in the object; it does not modify the value in the database.

setBillerID()

Specifies the biller ID.

Syntax	<code>public void setBillerID (String <i>billerID</i>) ;</code>
Parameters	<i>billerID</i> — A string that specifies the biller ID.
Discussion	Only changes the value of the biller ID field in the object; it does not modify the value in the database.

setCaseNum()

Specifies the case number.

Syntax	<code>public void setCaseNum (String <i>caseNum</i>) ;</code>
Parameters	<i>caseNum</i> — A string that specifies the case number.
Discussion	Only changes the value of the case number field in the object; it does not modify the value in the database.

setCsrName()

Specifies the CSR (Case Register Number) name.

Syntax	<code>public void setCsrName (String <i>csrName</i>) ;</code>
Parameters	<i>acctID</i> — A string that specifies the account ID.
Discussion	Only changes the value of the account ID field in the object; it does not modify the value in the database.

setDocID()

Specifies the document ID.

Syntax	<code>public void setExtension (String <i>docID</i>) ;</code>
Parameters	<i>docID</i> — A string that specifies the doc ID.
Discussion	Only changes the value of the doc ID field in the object; it does not modify the value in the database.

setExtension()

Specifies the messages extension.

Syntax	<code>public void setExtension (String <i>extension</i>) ;</code>
Parameters	<i>extension</i> — A string that specifies the extension.
Discussion	Only changes the value of the extension field in the object; it does not modify the value in the database.

setMessage()

Specifies the message contents.

Syntax	<code>public void setMessage (String <i>message</i>) ;</code>
Parameters	<i>message</i> — A string that specifies the message.
Discussion	Only changes the value of the message field in the object; it does not modify the value in the database.

setMsgFrom()

Specifies the sender.

Syntax	<code>public void setMsgFrom (String <i>msgFrom</i>) ;</code>
Parameters	<i>msgFrom</i> — A string that specifies the sender.
Discussion	Only changes the value of the message from field in the object; it does not modify the value in the database.

setMsgStatus()

Specifies the status.

Syntax	<code>public void setMsgStatus (String <i>msgStatus</i>) ;</code>
Parameters	<i>msgStatus</i> — A string that specifies the message status.
Discussion	Only changes the value of the message status field in the object; it does not modify the value in the database.

setMsgTo()

Specifies the receiver.

Syntax	<code>public void setMsgTo (String <i>msgTo</i>) ;</code>
Parameters	<i>msgTo</i> — A string that specifies the message recipient.
Discussion	Only changes the value of the message to field in the object; it does not modify the value in the database.

setSenderType()

Specifies the sender type.

Syntax	<code>public void setSenderType (String <i>senderType</i>) ;</code>
Parameters	<i>senderType</i> — A string that specifies the sender type.
Discussion	Only changes the value of the sender type field in the object; it does not modify the value in the database.

setSubject()

Specifies the subject.

Syntax	<code>public void setSubject (String <i>subject</i>) ;</code>
Parameters	<i>subject</i> — A string that specifies the subject.
Discussion	Only changes the value of the subject field in the object; it does not modify the value in the database.

setType()

Specifies the message type.

Syntax	<code>public void setType (String <i>type</i>) ;</code>
Parameters	<i>type</i> — A string that specifies the message type.
Discussion	Only changes the value of the message type field in the object; it does not modify the value in the database.

getExtDate1()

Retrieves the first extension date.

Syntax	<code>public Date getExtDate1();</code>

Parameters	None.

Discussion	

setExtDate1()

Specifies the first extension date.

Syntax	<code>public void setExtDate1(Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the first extension date.

Discussion	

getExtDate2()

Retrieves the second extension date.

Syntax	<code>public Date getExtDate2();</code>

Parameters	None.

Discussion	

setExtDate2()

Specifies the second extension date.

Syntax	<code>public void setExtDate2 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the second extension date.

Discussion	

getExtDate3()

Retrieves the third extension date.

Syntax	<code>public Date getExtDate3();</code>

Parameters	None.

Discussion

setExtDate3()

Specifies the third extension date.

Syntax	<code>public void setExtDate3 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the third extension date.

Discussion

getExtDate4()

Retrieves the fourth extension date.

Syntax	<code>public Date getExtDate4();</code>

Parameters	None.

Discussion

setExtDate4()

Specifies the fourth extension date.

Syntax	<code>public void setExtDate4 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the fourth extension date.

Discussion

getExtNum1()

Retrieves the first extension number.

Syntax	<code>public double getExtNum1();</code>

Parameters	None.

Discussion

setExtNum1()

Specifies the first extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the first extension number.
Discussion	

getExtNum2()

Retrieves the second extension number.

Syntax	<code>public double getExtNum2();</code>
Parameters	None.
Discussion	

setExtNum2()

Specifies the second extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the second extension number.
Discussion	

getExtNum3()

Retrieves the third extension number.

Syntax	<code>public double getExtNum3();</code>
Parameters	None.
Discussion	

setExtNum3()

Specifies the third extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>

Parameters	<i>num</i> — A double that indicates the third extension number.

Discussion

getExtNum4()

Retrieves the fourth extension number.

Syntax	<code>public double getExtNum4();</code>

Parameters	None.

Discussion

setExtNum4()

Specifies the fourth extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>

Parameters	<i>num</i> — A double that indicates the fourth extension number.

Discussion

BXOFXSignOn Class Reference

The BXOFXSignOn base class handles the Open Financial Exchange (OFX) sign on.

Whenever possible, use the interface class IBXOFXSignOn (see “IBXOFXSignOn Class Reference”) instead of BXOFXSignOn.

IBXOFXSignOn Class Reference

The IBXOFXSignOn interface class handles the Open Financial Exchange (OFX) sign on.

Class Description

Table 2-28 describes the IBXOFXSignOn interface.

Table 2-28 IBXOFXSignOn Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public interface IBXOFXSignOn extends IBXBillingEntity {...}</code>

Methods

This section describes the IBXOFXSignOn methods. Table 2-29 lists these methods.

Table 2-29 IBXOFXSignOn Class Methods

Method	Summary
<code>getPassword()</code>	Retrieves the OFX sign on password.
<code>setPassword()</code>	Specifies the OFX sign on password.
<code>getConsolidatorName()</code>	Retrieves the consolidator name.
<code>setConsolidatorName()</code>	Specifies the consolidator name.

getPassword()

Retrieves the password for the OFX sign on.

Syntax	<code>public String getPassword();</code>
Parameters	None.
Discussion	

setPassword()

Specifies the password for the OFX sign on.

Syntax	<code>public void setPassword(String value);</code>
Parameters	<i>value</i> — A string representing the password value.
Discussion	

setConsolidatorName()

Specifies the consolidator name.

Syntax	<code>public void setConsolidatorName(String value);</code>
Parameters	<i>value</i> — A string representing the consolidator name.
Discussion	

getConsolidatorName()

Retrieves the consolidator name.

Syntax	<code>public String getConsolidatorName();</code>
Parameters	None.
Discussion	

setConsolidatorName()

Specifies the consolidator name.

Syntax	<code>public void setConsolidatorName(String value);</code>
Parameters	<i>value</i> — A string representing the consolidator name.
Discussion	

BXPAYHistory Class Reference

The IBXPAYHistory class defines methods that manipulate objects representing payment history records.

Whenever possible, use the interface class IBXPAYHistory (see “IBXPAYHistory Class Reference”) instead of BXPAYHistory.

IBXPayHistory Class Reference

The IBXPayHistory class defines methods that manipulate objects representing payment history records. You call the `BXBillingMgr.createPayHistory()` method to create a payment history object.

Before you can call the `IBXEntity.ejbCreate()` method (see “IBXBillingEntity Class Reference,” on page 164) to create a new record associated with the object, you must call the `setDocID()`, `setPmtAmount()`, and `setPmtTxnStatus()` methods to set the document ID, payment amount, and transaction status, respectively.

Table 2-30 describes the class definition for the IBXPayHistory interface.

Table 2-30 IBXPayHistory Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing;</code>
Syntax	<code>public interface IBXPayHistory extends IBXBillingEntity {...}</code>

Methods

This section describes the IBXPayHistory class methods. Table 2-31 lists these methods.

Table 2-31 IBXPayHistory Class Methods (1 of 3)

Methods	Description
Getting and setting payment history information	
<code>getDocID()</code>	Retrieves the document ID.
<code>setDocID()</code>	Specifies the document ID.
<code>getAccountID()</code>	Retrieves the payment account ID.
<code>setAccountID()</code>	Specifies the payment account ID.
<code>getCreateDate()</code>	Retrieves the payment history’s creation date.
<code>setCreateDate()</code>	Specifies the payment history’s creation date.
<code>getModifyDate()</code>	Retrieves the payment history’s modification date.
<code>setModifyDate()</code>	Specifies the payment history’s modification date.
<code>getPmtAmount()</code>	Retrieves the payment amount.
<code>setPmtAmount()</code>	Specifies the payment amount.
<code>getPmtTxnStatus()</code>	Retrieves the payment transaction status.

Table 2-31 IBXPayHistory Class Methods (2 of 3)

Methods	Description
<code>setPmtTxnStatus()</code>	Specifies the payment transaction status.
<code>getPmtTxnID()</code>	Retrieves the payment transaction ID.
<code>setPmtTxnID()</code>	Specifies the payment transaction ID.
<code>getPmtMode()</code>	Retrieves the payment mode.
<code>setPmtMode()</code>	Specifies the payment mode.
<code>getFileID()</code>	Retrieves the file ID.
<code>setFileID()</code>	Specifies the file ID.
<code>getSchedulePmtDate()</code>	Retrieves the scheduled payment date.
<code>setSchedulePmtDate()</code>	Specifies the scheduled payment date.
<code>getPaymentType()</code>	Retrieves the payment type.
<code>setPaymentType()</code>	Specifies the payment type.
<code>getPaymentBrand()</code>	Retrieves the payment brand.
<code>setPaymentBrand()</code>	Specifies the payment brand.
<code>getExtension()</code>	Retrieves the payment history's extension.
<code>setExtension()</code>	Specifies the payment history's extension. Discussion
<code>getNumProcessDays()</code>	Retrieves the number of processed days
<code>setNumProcessDays()</code>	Specifies the number of processed days
<code>getServerTrnID1()</code>	Retrieves the server transaction ID 1
<code>setServerTrnID1()</code>	Specifies the server transaction ID 1
<code>getServerTrnID2()</code>	Retrieves the server transaction ID 2
<code>setServerTrnID2()</code>	Specifies the server transaction ID 2
<code>getPmtProcessDate()</code>	Retrieves the Payment Process Date
<code>getExtDate1</code>	Retrieves the first extension date
<code>setExtDate1</code>	Specifies the first extension date.
<code>getExtDate2()</code>	Retrieves the second extension date
<code>setExtDate2()</code>	Specifies the second extension date.
<code>getExtDate3()</code>	Retrieves the third extension date
<code>setExtDate3()</code>	Specifies the third extension date.
<code>getExtDate4()</code>	Retrieves the fourth extension date

Table 2-31 IBXPayHistory Class Methods (3 of 3)

Methods	Description
setExtDate4 ()	Specifies the fourth extension date.
getExtNum1 ()	Retrieves the first extension number.
setExtNum1 ()	Specifies the first extension number.
getExtNum2 ()	Retrieves the second extension number.
setExtNum2 ()	Specifies the second extension number.
getExtNum3 ()	Retrieves the third extension number.
setExtNum3 ()	Specifies the third extension number.
getExtNum4 ()	Retrieves the fourth extension number.
setExtNum4 ()	Specifies the fourth extension number.
setPmtProcessDate ()	Specifies the Payment Process Date

getAccountID()

Retrieves the payment account ID.

Syntax	<code>public String getAccountID();</code>
Parameters	None.
Discussion	Returns a string that contains the account ID.

getCreateDate()

Retrieves the payment history's creation date.

Syntax	<code>public Date getCreateDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the creation date.

getDocID()

Retrieves the document ID.

Syntax	<code>public String getDocID();</code>
Parameters	None.
Discussion	Returns a string that contains the document ID.

getExtension()

Retrieves the payment history's extension.

Syntax	<code>public String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the extension.

getFileID()

Retrieves the file ID.

Syntax	<code>public String getFileID();</code>
Parameters	None.
Discussion	Returns a string that contains the file ID.

getModifyDate()

Retrieves the payment history's modification date.

Syntax	<code>public Date getModifyDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the modification date.

getPaymentBrand()

Retrieves the payment brand.

Syntax	<code>public Date getPaymentBrand();</code>

Parameters	None.
Discussion	Returns a Date object that contains the payment brand.

getPaymentType()
Retrieves the payment type.

Syntax	<code>public String getPaymentType();</code>
Parameters	None.
Discussion	Returns a string that contains the payment type.

getPmtAmount()
Retrieves the payment amount.

Syntax	<code>public double getPmtAmount();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the amount.

getPmtMode()
Retrieves the payment mode.

Syntax	<code>public String getPmtMode();</code>
Parameters	None.
Discussion	Returns a string that contains the mode.

getPmtTxnID()
Retrieves the payment transaction ID.

Syntax	<code>public String getPmtTxnID();</code>
Parameters	None.
Discussion	Returns a string that contains the transaction ID.

getPmtTxnStatus()

Retrieves the payment transaction status.

Syntax	<code>public String getPmtTxnStatus();</code>
Parameters	None.
Discussion	Returns a string that contains the payment transaction status

getSchedulePmtDate()

Retrieves the scheduled payment date.

Syntax	<code>public Date getSchedulePmtDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the scheduled payment date.

getNumProcessDays()

Retrieves the number of process days.

Syntax	<code>public String getNumProcessDays();</code>
Parameters	None.
Discussion	Returns a string that contains the number of process days

getServerTrnID1()

Retrieves the server transaction ID.

Syntax	<code>public String getServerTrnID1();</code>
Parameters	None.
Discussion	Returns a string that contains the server transaction ID 1

getServerTrnID2()

Retrieves the server transaction ID.

Syntax	<code>public String getServerTrnID2();</code>

Parameters	None.
Discussion	Returns a string that contains the server transaction ID 2

getPmtProcessDate()

Retrieves the payment process date.

Syntax	<code>public String getPmtProcessDate();</code>
Parameters	None.
Discussion	Returns a string that contains the payment process date

setAccountID()

Specifies the payment account ID.

Syntax	<code>public void setPmtAcctID (String <i>pmtAcctID</i>);</code>
Parameters	<i>pmtAcctID</i> — A string that specifies the ID
Discussion	Only changes the value of the payment account ID field in the object; it does not modify the value in the database.

setCreateDate()

Specifies the creation date of payments.

Syntax	<code>public void setCreateDate (Date <i>createDate</i>);</code>
Parameters	<i>createDate</i> — A date that specifies the creation date
Discussion	Only changes the value of the create date ID field in the object; it does not modify the value in the database.

setDocID()

Specifies the document ID.

Syntax	<code>public void setDocID (String <i>docID</i>);</code>
Parameters	<i>docID</i> — A string that specifies the document ID

Discussion	Only changes the value of the document ID field in the object; it does not modify the value in the database.

setModifyDate()

Specifies the modification date of payments.

Syntax	<code>public void setModifyDate (String <i>modifydate</i>) ;</code>
Parameters	<i>modifydate</i> — A string that specifies modification date.
Discussion	Only changes the value of the modify date ID field in the object; it does not modify the value in the database.

setExtension()

Specifies the payment history's extension.

Syntax	<code>public void setExtension (String <i>extension</i>) ;</code>
Parameters	<i>extension</i> — A string that specifies the extension
Discussion	Only changes the value of the extension field in the object; it does not modify the value in the database.

setFileID()

Specifies the file ID.

Syntax	<code>public void setFileID (String <i>fileID</i>) ;</code>
Parameters	<i>fileID</i> — A string that specifies the file ID.
Discussion	Only changes the value of the file ID field in the object; it does not modify the value in the database.

setPaymentType()

Specifies the payment type.

Syntax	<code>public void setPaymentBrand (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the value brand.

Discussion	Only changes the value of the payment brand field in the object; it does not modify the value in the database.

setPaymentType()
Specifies the payment type.

Syntax	<code>public void setPmtType (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the payment type
Discussion	Only changes the value of the payment type field in the object; it does not modify the value in the database.

setPmtAmount()
Specifies the payment amount.

Syntax	<code>public void setPmtAmount (double <i>pmtAmount</i>) ;</code>
Parameters	<i>pmtAmount</i> — A double-precision floating point number that specifies the amount
Discussion	

setPmtMode()
Specifies the payment mode.

Syntax	<code>public void setPmtMode (String <i>pmtMode</i>) ;</code>
Parameters	<i>pmtMode</i> — A string that specifies the mode.
Discussion	Only changes the value of the payment mode field in the object; it does not modify the value in the database.

setPmtTxnID()
Specifies the payment transaction ID.

Syntax	<code>public void setPmtTxnID (String <i>pmtTxnID</i>) ;</code>
Parameters	<i>pmtTxnID</i> — A string that specifies the payment transaction ID.

Discussion	Only changes the value of the payment transaction ID field in the object; it does not modify the value in the database.

setPmtTxnStatus()

Specifies the payment transaction status.

Syntax	<code>public void setPmtStatus (string <i>pmtTxnStatus</i>) ;</code>
Parameters	<i>pmtTxnStatus</i> — A string that specifies the status.
Discussion	Only changes the value of the payment status field in the object; it does not modify the value in the database.

setNumProcessDays()

Specifies the number of process days on payments.

Syntax	<code>public void setNumProcessDays (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the number of process days.
Discussion	Only changes the value of the value field in the object; it does not modify the value in the database.

setServerTrnID1()

Specifies the server transaction ID 1.

Syntax	<code>public void setSeverTrnID1 (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that species the server transaction ID 1.
Discussion	Only changes the value of the value field in the object; it does not modify the value in the database.

setServerTrnID2()

Specifies the server transaction ID 2.

Syntax	<code>public void setSeverTrnID2 (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the server transaction ID 2.

Discussion	Only changes the value of the value field in the object; it does not modify the value in the database.

setPmtProcessDate()
Specifies the payment process date.

Syntax	<code>public void setPmtProcessDate (Date <i>value</i>) ;</code>
Parameters	<i>value</i> — Specifies the payment process date
Discussion	Only changes the value of the value field in the object; it does not modify the value in the database.

getExtDate1()
Retrieves the first extension date.

Syntax	<code>public Date getExtDate1();</code>
Parameters	None.
Discussion	

setExtDate1()
Specifies the first extension date.

Syntax	<code>public void setExtDate1(Date <i>date</i>) ;</code>
Parameters	<i>date</i> — A date that indicates the first extension date.
Discussion	

getExtDate2()
Retrieves the second extension date.

Syntax	<code>public Date getExtDate2();</code>
Parameters	None.
Discussion	

setExtDate2()

Specifies the second extension date.

Syntax	<code>public void setExtDate2 (Date <i>date</i>);</code>
Parameters	<i>date</i> — A date that indicates the second extension date.
Discussion	

getExtDate3()

Retrieves the third extension date.

Syntax	<code>public Date getExtDate3();</code>
Parameters	None.
Discussion	

setExtDate3()

Specifies the third extension date.

Syntax	<code>public void setExtDate3 (Date <i>date</i>);</code>
Parameters	<i>date</i> — A date that indicates the third extension date.
Discussion	

getExtDate4()

Retrieves the fourth extension date.

Syntax	<code>public Date getExtDate4();</code>
Parameters	None.
Discussion	

setExtDate4()

Specifies the fourth extension date.

Syntax	<code>public void setExtDate4 (Date <i>date</i>);</code>

Parameters	date — A date that indicates the fourth extension date.
Discussion	

getExtNum1()
Retrieves the first extension number.

Syntax	public double getExtNum1 ();
Parameters	None.
Discussion	

setExtNum1()
Specifies the first extension number.

Syntax	public void setExtNum1 (double num);
Parameters	num — A double that indicates the first extension number.
Discussion	

getExtNum2()
Retrieves the second extension number.

Syntax	public double getExtNum2 ();
Parameters	None.
Discussion	

setExtNum2()
Specifies the second extension number.

Syntax	public void setExtNum1 (double num);
Parameters	num — A double that indicates the second extension number.
Discussion	

getExtNum3()

Retrieves the third extension number.

Syntax	<code>public double getExtNum3 ();</code>
Parameters	None.
Discussion	

setExtNum3()

Specifies the third extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the third extension number.
Discussion	

getExtNum4()

Retrieves the fourth extension number.

Syntax	<code>public double getExtNum4 ();</code>
Parameters	None.
Discussion	

setExtNum4()

Specifies the fourth extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the fourth extension number.
Discussion	

BXPaySchedule Class Reference

The BXPaySchedule base class defines methods that manipulate objects representing payment schedule records.

Whenever possible, use the interface class IBXPaySchedule (see “IBXPaySchedule Class Reference) instead of BXPaySchedule.

IBXPaySchedule Class Reference

The IBXPaySchedule interface defines methods that manipulate objects representing payment schedule records. You call the `BXBillingMgr.createPaySchedule()` method to create a payment schedule object.

Before you can call the `IBXEntity.ejbCreate()` (see “IBXBillingEntity Class Reference,” on page 164) method to create a new record associated with the object, you must call the `setAccountID()` method to set the account ID.

Class Definition

Table 2-32 describes the IBXPaySchedule interface definition.

Table 2-32 IBXPaySchedule Class Definition

Package	com.iplanet.ecomm.billxb2b.ejb.billing
Syntax	public interface IBXPaySchedule extends IBXBillingEntity {...}

Methods

This section describes the IBXPaySchedule methods. Table 2-33 lists these methods.

Table 2-33 IBXPaySchedule Class Methods (1 of 3)

Methods	Description
<code>getAccountID()</code>	Retrieves the account ID.
<code>setAccountID()</code>	Specifies the account ID.
<code>getUserAccountID()</code>	Retrieves the user account ID.
<code>setUserAccountID()</code>	Specifies the user account ID.
<code>getBillerID()</code>	Retrieves the Biller ID.
<code>setBillerAcctNum()</code>	Specifies the Biller account number.

Table 2-33 IBXPaySchedule Class Methods (2 of 3)

Methods	Description
<code>getBillerAcctNum()</code>	Retrieves the Biller account number.
<code>setBillerID()</code>	Specifies the Biller ID.
<code>getStartOnDate()</code>	Retrieves the schedule start date.
<code>setStartOnDate()</code>	Specifies the schedule start date.
<code>getStopOnDate()</code>	Retrieves the schedule end date.
<code>setStopOnDate()</code>	Specifies the schedule end date.
<code>getFrequency()</code>	Retrieves the payment frequency.
<code>setFrequency()</code>	Specifies the payment frequency.
<code>getPayMode()</code>	Retrieves the payment mode.
<code>setPayMode()</code>	Specifies the payment mode.
<code>getDayBeforeDue()</code>	Retrieves the number of days before a payment is due.
<code>setDayBeforeDue()</code>	Specifies the number of days before a payment is due.
<code>getDayOfMonth()</code>	Retrieves the day of the month.
<code>setDayOfMonth()</code>	Specifies the day of the month.
<code>getPayDate()</code>	Retrieves the payment date.
<code>setPayDate()</code>	Specifies the payment date.
<code>getPayAmount()</code>	Retrieves the scheduled payment amount.
<code>setPayAmount()</code>	Specifies the scheduled payment amount.
<code>getPayAmountThreshold()</code>	Retrieves the type of the payment threshold.
<code>setPayAmountThreshold()</code>	Specifies the type of the payment threshold.
<code>getPayAmountType()</code>	Retrieves the type of the payment amount.
<code>setPayAmountType()</code>	Specifies the type of the payment amount.
<code>getPmtAcctID()</code>	Retrieves the payment account ID.
<code>setPmtAcctID()</code>	Specifies the payment account ID.
<code>getScheduleStatus()</code>	Retrieves the schedule status.
<code>setScheduleStatus()</code>	Specifies the schedule status.
<code>getExtDate1</code>	Retrieves the first extension date
<code>setExtDate1</code>	Specifies the first extension date.
<code>getExtDate2()</code>	Retrieves the second extension date

Table 2-33 IBXPaySchedule Class Methods (3 of 3)

Methods	Description
setExtDate2 ()	Specifies the second extension date.
getExtDate3 ()	Retrieves the third extension date
setExtDate3 ()	Specifies the third extension date.
getExtDate4 ()	Retrieves the fourth extension date
setExtDate4 ()	Specifies the fourth extension date.
getExtNum1 ()	Retrieves the first extension number.
setExtNum1 ()	Specifies the first extension number.
getExtNum2 ()	Retrieves the second extension number.
setExtNum2 ()	Specifies the second extension number.
getExtNum3 ()	Retrieves the third extension number.
setExtNum3 ()	Specifies the third extension number.
getExtNum4 ()	Retrieves the fourth extension number.
setExtNum4 ()	Specifies the fourth extension number.

getAccountID()

Retrieves the account ID.

Syntax	<code>public String getAccountID (String <i>accountID</i>) ;</code>
Parameters	<i>accountID</i> — A string that specifies the account ID.
Discussion	Returns a string that contains the ID.

getUserAccountID()

Retrieves the account ID.

Syntax	<code>public String getUserAccountID (String <i>accountID</i>) ;</code>
Parameters	<i>accountID</i> — A string that specifies the account ID.
Discussion	Returns a string that contains the user account ID.

getBillerID()

Retrieves the biller ID.

Syntax	<code>public String getBillerID (String <i>billerID</i>) ;</code>
Parameters	<i>billerID</i> — A string that specifies the biller ID.
Discussion	Returns a string that contains the biller ID.

getBillerAcctNum()

Retrieves the payment frequency.

Syntax	<code>public String getBillerAcctNum() ;</code>
Parameters	None.
Discussion	Returns a string that contains the Biller account number.

getFrequency()

Retrieves the payment frequency.

Syntax	<code>public String getFrequency() ;</code>
Parameters	None.
Discussion	Returns a string that contains the frequency.

getPayMode()

Retrieves the payment mode.

Syntax	<code>public String getPayMode () ;</code>
Parameters	None.
Discussion	Returns a string that contains the payment mode.

getDayBeforeDue()

Retrieves the day before due payment date.

Syntax	<code>public int getDayBeforeDue () ;</code>

Parameters	None.
Discussion	Returns a string that contains the day before payment due.

getDayOfMonth()
Retrieves the day of the month.

Syntax	<code>public int getDayOfMonth ();</code>
Parameters	None.
Discussion	Returns a string that contains the day of the month.

getPayDate()
Retrieves the payment date.

Syntax	<code>public Date getPayMode ();</code>
Parameters	None.
Discussion	It contains the payment date.

getPayAmount()
Retrieves the scheduled payment amount.

Syntax	<code>public double getPayAmount();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the amount.

getPayAmountThreshold()
Retrieves the scheduled payment amount.

Syntax	<code>public double getPayAmountThreshold();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the amount threshold.

getPayAmountType()

Retrieves the type of the payment amount.

Syntax	<code>public String getPayAmountType();</code>
Parameters	None.
Discussion	Returns a string that contains the type.

getPmtAcctID()

Retrieves the payment account ID.

Syntax	<code>public String getPmtAcctID();</code>
Parameters	None.
Discussion	Returns a string that contains the account ID.

getScheduleStatus()

Retrieves the schedule status.

Syntax	<code>public String getScheduleStatus();</code>
Parameters	None.
Discussion	Returns a string that contains the status.

getStartOnDate()

Retrieves the schedule start date.

Syntax	<code>public void setStartOnDate (Date <i>startDate</i>);</code>
Parameters	<i>startDate</i> — A date that specifies the start date.
Discussion	Returns a Date object that contains the start date.

getStopOnDate()

Retrieves the schedule end date.

Syntax	<code>public Date getStopOnDate();</code>

Parameters	None.
Discussion	Returns a Date object that contains the end date.

setAccountID()
Specifies the account ID.

Syntax	<code>public void setAccountID (String <i>accountID</i>) ;</code>
Parameters	<i>accountID</i> — A string that specifies the account ID
Discussion	Only changes the value of the account ID field in the object; it does not modify the value in the database.

setUserAccountID()
Specifies the account ID.

Syntax	<code>public void setUserAccountID (String <i>accountID</i>) ;</code>
Parameters	<i>accountID</i> — A string that specifies the ID
Discussion	Only changes the value of the user account ID field in the object; it does not modify the value in the database.

setBillerID()
Specifies the Biller ID.

Syntax	<code>public void setBillerID (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the biller ID
Discussion	Only changes the value of the billerID in the object; it does not modify the value in the database.

setBillerAcctNum()
Specifies the biller account number.

Syntax	<code>public void setAccountID (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the biller account number.

Discussion	Only changes the value of the biller account ID field in the object; it does not modify the value in the database.

setDayBeforeDue()

Specifies the number of days before a payment is due.

Syntax	<code>public void setDayBeforeDue (int <i>dayBeforeDue</i>) ;</code>
Parameters	<i>dayBeforeDue</i> — An integer that specifies the number of days.
Discussion	Only changes the value of the day before due field in the object; it does not modify the value in the database.

setFrequency()

Specifies the payment frequency.

Syntax	<code>public void setFrequency (String <i>frequency</i>) ;</code>
Parameters	<i>frequency</i> — A string that specifies the frequency
Discussion	Only changes the value of the frequency field in the object; it does not modify the value in the database.

setPayMode()

Specifies the payment mode.

Syntax	<code>public void setPayMode (String <i>payMode</i>) ;</code>
Parameters	<i>payMode</i> — A string that specifies the payment mode
Discussion	Only changes the value of the payment mode field in the object; it does not modify the value in the database.

setPayDate()

Specifies the payment mode.

Syntax	<code>public void setPayDate (Date <i>payDate</i>) ;</code>
Parameters	<i>payDate</i> — Specifies the pay date

Discussion	Only changes the value of the pay date field in the object; it does not modify the value in the database.

setDayOfMonth()
Specifies the day of the month.

Syntax	<code>public void setDayOfMonth (int <i>dayOfMonth</i>) ;</code>
Parameters	<i>dayOfMonth</i> — An integer that specifies the day of the month
Discussion	Only changes the value of the day of month field in the object; it does not modify the value in the database.

setPayAmount()
Specifies the scheduled payment amount.

Syntax	<code>public void setAmount (double <i>payAmount</i>) ;</code>
Parameters	<i>payAmount</i> — A double-precision floating point number that specifies the amount.
Discussion	Only changes the value of the amount field in the object; it does not modify the value in the database.

setPayAmountType()
Specifies the type of the payment amount.

Syntax	<code>public void setPayAmountType (String <i>payAmountType</i>) ;</code>
Parameters	<i>payAmountType</i> — A string that specifies the type.
Discussion	Only changes the value of the pay amount type field in the object; it does not modify the value in the database.

setPayAmountThreshold()
Specifies the type of the payment amount threshold.

Syntax	<code>public void setPayAmountThreshold (double <i>payAmountThreshold</i>) ;</code>

Parameters	<i>payAmountThreshold</i> — A double precision floating point number that Specifies the threshold.
Discussion	Only changes the value of the pay amount threshold type field in the object; it does not modify the value in the database.

setPmtAcctID()

Specifies the payment account ID.

Syntax	<code>public void setPmtAcctID (String <i>pmtAcctID</i>) ;</code>
Parameters	<i>pmtAcctID</i> — A string that specifies the ID.
Discussion	Only changes the value of the payment account ID field in the object; it does not modify the value in the database.

setScheduleStatus()

Specifies the schedule status.

Syntax	<code>public void setScheduleStatus (String <i>status</i>);</code>
Parameters	<i>status</i> — A string that specifies the status.
Discussion	Only changes the value of the status field in the object; it does not modify the value in the database.

setStartDate()

Specifies the schedule start date.

Syntax	<code>public void setStartDate (Date <i>startDate</i>) ;</code>
Parameters	<i>startDate</i> — A Date that specifies the start date.
Discussion	Only changes the value of the start date field in the object; it does not modify the value in the database.

setStopOnDate()

Specifies the schedule end date.

Syntax	<code>public void setStopOnDate (Date <i>stopDate</i>) ;</code>

Parameters	<i>stopDate</i> — A Date that specifies the end date
Discussion	Only changes the value of the end date field in the object; it does not modify the value in the database.

getExtDate1()
Retrieves the first extension date.

Syntax	<code>public Date getExtDate1();</code>
Parameters	None.
Discussion	

setExtDate1()
Specifies the first extension date.

Syntax	<code>public void setExtDate1(Date date);</code>
Parameters	<i>date</i> — A date that indicates the first extension date.
Discussion	

getExtDate2()
Retrieves the second extension date.

Syntax	<code>public Date getExtDate2();</code>
Parameters	None.
Discussion	

setExtDate2()
Specifies the second extension date.

Syntax	<code>public void setExtDate2 (Date date);</code>
Parameters	<i>date</i> — A date that indicates the second extension date.
Discussion	

getExtDate3()

Retrieves the third extension date.

Syntax	<code>public Date getExtDate3();</code>

Parameters	None.

Discussion	

setExtDate3()

Specifies the third extension date.

Syntax	<code>public void setExtDate3 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the third extension date.

Discussion	

getExtDate4()

Retrieves the fourth extension date.

Syntax	<code>public Date getExtDate4();</code>

Parameters	None.

Discussion	

setExtDate4()

Specifies the fourth extension date.

Syntax	<code>public void setExtDate4 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the fourth extension date.

Discussion	

getExtNum1()

Retrieves the first extension number.

Syntax	<code>public double getExtNum1();</code>

Parameters	None.
Discussion	

setExtNum1()
Specifies the first extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>) ;</code>
Parameters	<i>num</i> — A double that indicates the first extension number.
Discussion	

getExtNum2()
Retrieves the second extension number.

Syntax	<code>public double getExtNum2 () ;</code>
Parameters	None.
Discussion	

setExtNum2()
Specifies the second extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>) ;</code>
Parameters	<i>num</i> — A double that indicates the second extension number.
Discussion	

getExtNum3()
Retrieves the third extension number.

Syntax	<code>public double getExtNum3 () ;</code>
Parameters	None.
Discussion	

setExtNum3()

Specifies the third extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the third extension number.
Discussion	

getExtNum4()

Retrieves the fourth extension number.

Syntax	<code>public double getExtNum4 ();</code>
Parameters	None.
Discussion	

setExtNum4()

Specifies the fourth extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the fourth extension number.
Discussion	

BXSummary Class Reference

The BXSummary base class defines methods that manipulate objects representing billing summary records.

Whenever possible, use the interface class IBXSummary (see “IBXSummary Class Reference”) instead of BXSummary.

IBXSummary Class Reference

The IBXSummary interface class defines methods that manipulate objects representing billing summary records. You call the `BXBillingMgr.createSummary()` method to create Returns a summary object.

Before you can call the `IBXEntity.ejbCreate()` (see “IBXBillingEntity Class Reference,” on page 164) method to create a new record associated with the object, you must call the `setdocID()`, `setStartDate()`, `setEndDate()`, and `setDueDate()` methods to set the document ID, and summary start, end, and due dates, respectively.

Class Definition

Table 2-34 describes the IBXSummary class interface definition.

Table 2-34 IBXSummary Class Definition	
Package	<code>com.iplanet.ecomm.billxb2b.ejb.billing</code>
Syntax	<code>public interface IBXSummary extends IBXBillingEntity {...}</code>

Methods

This section describes the IBXSummary class methods. Table 2-35 lists these methods.

Table 2-35 IBXSummary Class Methods (1 of 4)

Methods	Description
Manipulating summary-related records	
<code>ejbFindByDocumentID()</code>	Retrieves the summary associated with this document.
<code>ejbFindByContainerID()</code>	Retrieves the summary associated with this container.
Getting and setting summary information	
<code>getContainerID()</code>	Retrieves the summary’s container ID.
<code>setContainerID()</code>	Specifies the summary’s container ID.
<code>getBillerID()</code>	Retrieves the Biller ID.
<code>setBillerID()</code>	Specifies the Biller ID.
<code>getDocID()</code>	Retrieves the summary’s document ID.

Table 2-35 IBXSummary Class Methods (2 of 4)

Methods	Description
<code>setDocID()</code>	Specifies the summary's document ID.
<code>getFileID()</code>	Retrieves the file ID.
<code>setFileID()</code>	Specifies the file ID.
<code>getAutoPayCalculated()</code>	Retrieves the autopay calculation.
<code>setAutoPayCalculated()</code>	Specifies the autopay calculation.
<code>getStartDate()</code>	Retrieves the starting date.
<code>setStartDate()</code>	Specifies the starting date.
<code>getEndDate()</code>	Retrieves the ending date.
<code>setEndDate()</code>	Specifies the ending date.
<code>getOriginalAmount()</code>	Retrieves the original amount.
<code>setOriginalAmount()</code>	Specifies the original amount.
<code>getCurrentBalance()</code>	Retrieves the current balance.
<code>setCurrentBalance()</code>	Specifies the current balance.
<code>getCurrentCharge()</code>	Retrieves the current charge.
<code>setCurrentCharge()</code>	Specifies the current charge.
<code>getPreviousBalance()</code>	Retrieves the previous balance.
<code>setPreviousBalance()</code>	Specifies the previous balance.
<code>getPreviousPayment()</code>	Retrieves the previous payment.
<code>setPreviousPayment()</code>	Specifies the previous payment.
<code>getBalanceDate()</code>	Retrieves the balance date.
<code>setBalanceDate()</code>	Specifies the balance date.
<code>getDueDate()</code>	Retrieves the due date.
<code>setDueDate()</code>	Specifies the due date.
<code>getLateAfterDate()</code>	Retrieves the late-after date.
<code>setLateAfterDate()</code>	Specifies the late-after date.
<code>getPenaltyType()</code>	Retrieves the penalty type.
<code>setPenaltyType()</code>	Specifies the penalty type.
<code>getPenaltyRate()</code>	Retrieves the penalty rate.
<code>setPenaltyRate()</code>	Specifies the penalty rate.

Table 2-35 IBXSummary Class Methods (3 of 4)

Methods	Description
<code>getMinPayment ()</code>	Retrieves the minimum payment.
<code>setMinPayment ()</code>	Specifies the minimum payment.
<code>getPaymentAmount ()</code>	Retrieves the payment amount.
<code>setPaymentAmount ()</code>	Specifies the payment amount.
<code>getLastPayDate ()</code>	Retrieves the last payment date.
<code>setLastPayDate ()</code>	Specifies the last payment date.
<code>getPayStatus ()</code>	Retrieves the payment status.
<code>setPayStatus ()</code>	Specifies the payment status.
<code>getSumStatus ()</code>	Retrieves the summary status.
<code>setSumStatus ()</code>	Specifies the summary status.
<code>getPayterms ()</code>	Retrieves the payment terms code.
<code>setPayterms ()</code>	Specifies the payment terms code.
<code>getDiscountBeforeDate ()</code>	Retrieves the discount-before date.
<code>setDiscountBeforeDate ()</code>	Specifies the discount-before date.
<code>getDiscountType ()</code>	Retrieves the discount type.
<code>setDiscountType ()</code>	Specifies the discount type.
<code>getExtDate1</code>	Retrieves the first extension date
<code>setExtDate1</code>	Specifies the first extension date.
<code>getExtDate2 ()</code>	Retrieves the second extension date
<code>setExtDate2 ()</code>	Specifies the second extension date.
<code>getExtDate3 ()</code>	Retrieves the third extension date
<code>setExtDate3 ()</code>	Specifies the third extension date.
<code>getExtDate4 ()</code>	Retrieves the fourth extension date
<code>setExtDate4 ()</code>	Specifies the fourth extension date.
<code>getExtNum1 ()</code>	Retrieves the first extension number.
<code>setExtNum1 ()</code>	Specifies the first extension number.
<code>getExtNum2 ()</code>	Retrieves the second extension number.
<code>setExtNum2 ()</code>	Specifies the second extension number.
<code>getExtNum3 ()</code>	Retrieves the third extension number.

Table 2-35 IBXSummary Class Methods (4 of 4)

Methods	Description
<code>setExtNum3()</code>	Specifies the third extension number.
<code>getExtNum4()</code>	Retrieves the fourth extension number.
<code>setExtNum4()</code>	Specifies the fourth extension number.

`ejbFindByContainerID()`

Retrieves the summary associated with this container.

Syntax	<code>public void ejbFindByContainerID (String key) throws Exception;</code>
Parameters	<i>key</i> — A string that specifies the container ID.
Discussion	The <i>key</i> parameter specifies the record to read. This method throws an exception if the record with this key does not exist. The fields of this object are set by from the contents of the record. The database context must have been set by calling the <code>createBillingContext()</code> method.

`ejbFindByDocumentID()`

Retrieves the summary associated with this document.

Syntax	<code>public void ejbFindByDocumentID (String key) throws Exception;</code>
Parameters	<i>key</i> — A string that specifies the document ID
Discussion	The <i>key</i> parameter specifies the record to read. This method throws an exception if the record with this key does not exist. The fields of this object are set by from the contents of the record. The database context must have been set by calling the <code>createBillingContext()</code> method.

`getBalanceDate()`

Retrieves the balance date.

Syntax	<code>public Date getBalanceDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the balance date.

getContainerID()

Retrieves the summary’s container ID.

Syntax	<code>public String getContainerID();</code>
Parameters	None.
Discussion	Returns a string that contains the ID.

getBillerID()

Retrieves the biller identification.

Syntax	<code>public String getBillerID();</code>
Parameters	None.
Discussion	Returns a string that contains the Biller ID.

getCurrentBalance()

Retrieves the current balance.

Syntax	<code>public double getCurrentBalance();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the balance.

getCurrentCharge()

Retrieves the current charge.

Syntax	<code>public double getCurrentCharge();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the amount.

getDiscountBeforeDate()

Retrieves the discount-before date.

Syntax	<code>public Date getDiscountBeforeDate();</code>

Parameters	None.
Discussion	Returns a Date object that contains the date.

getDescription()

Retrieves the summary's description.

Syntax	<code>public String getDescription();</code>
Parameters	None.
Discussion	Returns a string that contains the description.

getDiscountRate()

Retrieves the discount rate.

Syntax	<code>public double getDiscountRate();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the rate.

getDiscountType()

Retrieves the discount type.

Syntax	<code>public String getDiscountType();</code>
Parameters	None.
Discussion	Returns a string that contains the discount type.

getDocID()

Retrieves the summary's document ID.

Syntax	<code>public String getDocID;</code>
Parameters	None.
Discussion	Returns a string that contains the document ID.

getDueDate()

Retrieves the due date.

Syntax	<code>public Date getDueDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the due date.

getEndDate()

Retrieves the ending date.

Syntax	<code>public Date getEndDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the date.

getExtension()

Retrieves the summary’s extension.

Syntax	<code>public String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the extension.

getFileID()

Retrieves the file ID.

Syntax	<code>public String getFileID();</code>
Parameters	None.
Discussion	Returns a string that contains the file ID.

getAutoPayCalculated()

Retrieves the autopay calculation.

Syntax	<code>public String getAutoPayCalculated();</code>

Parameters	None.
Discussion	Returns a string that contains the autopay calculation.

getLastPayDate()

Retrieves the last payment date.

Syntax	<code>public Date getLastPayDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the last payment date.

getLateAfterDate()

Retrieves the date after which the payment is late.

Syntax	<code>public Date getLateAfterDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the date after which the payment is late.

getMinPayment()

Retrieves the minimum payment.

Syntax	<code>public double getMinPayment();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the minimum payment amount.

getOriginalAmount()

Retrieves the original amount.

Syntax	<code>public double getOriginalAmount();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the original amount.

getPayStatus()

Retrieves the payment status.

Syntax	<code>public String getPayStatus();</code>
Parameters	None.
Discussion	Returns a string that contains the payment status.

getPaymentAmount()

Retrieves the payment amount.

Syntax	<code>public double getPaymentAmount();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the payment amount.

getPayterms()

Retrieves the payment terms code.

Syntax	<code>public String getPayterms();</code>
Parameters	None.
Discussion	Returns a string that contains the code.

getPenaltyRate()

Retrieves the penalty rate.

Syntax	<code>public double getPenaltyRate();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the penalty rate.

getPenaltyType()

Retrieves the penalty type.

Syntax	<code>public String getPenaltyType();</code>
Parameters	None.
Discussion	Returns a string that contains the penalty type.

getPreviousBalance()

Retrieves the previous balance.

Syntax	<code>public double getPreviousBalance();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the previous balance.

getPreviousPayment()

Retrieves the previous payment.

Syntax	<code>public double getPreviousPayment();</code>
Parameters	None.
Discussion	Returns a double-precision floating point number that contains the previous payment amount.

getStartDate()

Retrieves the starting date.

Syntax	<code>public Date getStartDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the starting date.

getSumStatus()

Retrieves the summary status.

Syntax	<code>public String getSumStatus();</code>
Parameters	None.
Discussion	Returns a string that contains the summary status.

getURL()

Retrieves the URL.

Syntax	<code>public String getURL();</code>
Parameters	None.
Discussion	Returns a string that contains the URL.

setBalanceDate()

Specifies the balance date.

Syntax	<code>public void setBalanceDate(Date <i>date</i>);</code>
Parameters	<i>date</i> — A Date object that specifies the balance date.
Discussion	Only changes the value of the balance date in the object; it does not modify the value in the database.

setContainerID()

Specifies the summary's container ID.

Syntax	<code>public void setContainerID(String <i>containerID</i>);</code>
Parameters	<i>containerID</i> — A string that specifies the summary's container ID.
Discussion	Only changes the value of the container ID in the object; it does not modify the value in the database.

setBillerID()

Specifies the Biller ID.

Syntax	<code>public void setBillerID (String <i>billerID</i>);</code>
Parameters	<i>billerID</i> — A string that specifies the biller ID
Discussion	Only changes the value of the billerID in the object; it does not modify the value in the database.

setCurrentBalance()

Specifies the current balance.

Syntax	<code>public void setCurrentBalance(double <i>amount</i>);</code>
Parameters	<i>amount</i> — A double-precision floating point number that specifies the amount
Discussion	Only changes the value of the current balance in the object; it does not modify the value in the database.

setCurrentCharge()

Specifies the current charge.

Syntax	<code>public void setCurrentCharge(double <i>amount</i>);</code>
Parameters	<i>amount</i> — A double-precision floating point number that specifies the amount
Discussion	Only changes the value of the current charge in the object; it does not modify the value in the database.

setDescription()

Specifies the summary's description.

Syntax	<code>public void setDescription(String <i>description</i>);</code>
Parameters	<i>description</i> — A string that specifies the description.
Discussion	Only changes the value of the description in the object; it does not modify the value in the database.

setDiscountBeforeDate()

Specifies the discount-before date.

Syntax	<code>public void setBalanceDate(Date <i>date</i>);</code>
Parameters	<i>date</i> — A Date object that specifies the date
Discussion	Only changes the value of the balance date in the object; it does not modify the value in the database.

setDiscountRate()

Specifies the discount rate.

Syntax	<code>public void setDiscountRate(double <i>amount</i>);</code>
Parameters	<i>amount</i> — A double-precision floating point number that specifies the amount
Discussion	Only changes the value of the amount in the object; it does not modify the value in the database.

setDiscountType()

Specifies the dispute reason.

Syntax	<code>public void setDiscountType(String <i>type</i>);</code>
Parameters	<i>type</i> — A string that specifies the type
Discussion	Only changes the value of the type in the object; it does not modify the value in the database.

setDocID()

Specifies the summary's document ID.

Syntax	<code>public void setDocID(String <i>docID</i>);</code>
Parameters	<i>docID</i> — A string that specifies the document ID.
Discussion	Only changes the value of the doc ID in the object; it does not modify the value in the database.

setDueDate()

Specifies the due date.

Syntax	<code>public void setDueDate(Date <i>date</i>);</code>
Parameters	<i>date</i> — A string that specifies the date.
Discussion	Only changes the value of the date in the object; it does not modify the value in the database.

setEndDate()

Specifies the ending date.

Syntax	<code>public void setEndDate (Date <i>date</i>);</code>
Parameters	<i>date</i> — A Date object that specifies the date.
Discussion	Only changes the value of the ending date in the object; it does not modify the value in the database.

setExtension()

Specifies the summary's extension.

Syntax	<code>public void setExtension(String <i>extension</i>);</code>
Parameters	<i>extension</i> — A string that specifies the extension
Discussion	Only changes the value of the extension field in the object; it does not modify the value in the database.

setFileID()

Specifies the file ID.

Syntax	<code>public void setFileID(String <i>fileID</i>);</code>
Parameters	<i>fileID</i> — A string that specifies the ID.
Discussion	Only changes the value of the file ID in the object; it does not modify the value in the database.

setAutoPayCalculated()

Specifies the automatic payment calculation.

Syntax	<code>public void setAutoPayCalculated (String <i>autoPayCalculated</i>) ;</code>
Parameters	<i>autoPayCalculated</i> — A string that specifies the automatic pay calculation.
Discussion	Only changes the value of the automatic pay calculated in the object; it does not modify the value in the database.

setLastPayDate()

Specifies the last payment date.

Syntax	<code>public void setLastPayDate (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — A date object that specifies the last payment date.
Discussion	Only changes the value of the date in the object; it does not modify the value in the database.

setLateAfterDate()

Specifies the date after which the payment is late.

Syntax	<code>public void setLateAfterDate (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — A date object that specifies the date after which the payment is late.
Discussion	Only changes the value of the late date in the object; it does not modify the value in the database.

setMinPayment()

Specifies the minimum payment.

Syntax	<code>public void setMinPayment (double <i>amount</i>) ;</code>
Parameters	<i>amount</i> — A double-precision floating point number that specifies the minimum payment amount.
Discussion	Only changes the value of the amount in the object; it does not modify the value in the database.

setOriginalAmount()

Specifies the original amount.

Syntax	<code>public void setOriginalAmount(double <i>amount</i>);</code>
Parameters	<i>amount</i> — A double-precision floating point number that specifies the original amount.
Discussion	Only changes the value of the amount in the object; it does not modify the value in the database.

setPayStatus()

Specifies the payment status.

Syntax	<code>public void setPayStatus(String <i>status</i>);</code>
Parameters	<i>status</i> — A string that specifies the status.
Discussion	Only changes the value of the payment status in the object; it does not modify the value in the database.

setPaymentAmount()

Specifies the original amount.

Syntax	<code>public void setPaymentAmount(double <i>amount</i>);</code>
Parameters	<i>amount</i> — A double-precision floating point number that specifies the amount.
Discussion	Only changes the value of the amount in the object; it does not modify the value in the database.

setPayterms()

Specifies the payment terms code.

Syntax	<code>public void set PayTerms(String <i>code</i>);</code>
Parameters	<i>code</i> — A string that specifies the code.
Discussion	Only changes the value of the payment terms code in the object; it does not modify the value in the database.

setPenaltyRate()

Specifies the penalty rate.

Syntax	<code>public void setPenaltyRate(double <i>amount</i>);</code>
Parameters	<i>amount</i> — A double precision floating point number that specifies the penalty rate.
Discussion	Only changes the value of the penalty rate in the object; it does not modify the value in the database.

setPenaltyType()

Specifies the penalty type.

Syntax	<code>public void setPenaltyType(String <i>penaltyType</i>);</code>
Parameters	<i>penaltyType</i> — A string that specifies the penalty type.
Discussion	Only changes the value of the penalty type in the object; it does not modify the value in the database.

setPreviousBalance()

Displays previous balance.

Syntax	<code>public void setPreviousBalance(double <i>amount</i>);</code>
Parameters	<i>amount</i> — A double precision floating number that specifies the amount.
Discussion	Only changes the value of the previous payment in the object; it does not modify the value in the database.

setStartDate()

Specifies the starting date.

Syntax	<code>public void setStartDate(Date <i>date</i>);</code>
Parameters	<i>date</i> — A Date object that specifies the start date.
Discussion	Only changes the value of the start date in the object; it does not modify the value in the database.

setSumStatus()

Specifies the summary status.

Syntax	<code>public void setSumStatus(String <i>status</i>);</code>
Parameters	<i>status</i> — A string that specifies the summary status.
Discussion	Only changes the value of the summary status in the object; it does not modify the value in the database.

setURL()

Specifies the URL.

Syntax	<code>public void setURL(String <i>url</i>);</code>
Parameters	<i>url</i> — A string that specifies the URL.
Discussion	Only changes the value of the URL in the object; it does not modify the value in the database.

getExtDate1()

Retrieves the first extension date.

Syntax	<code>public Date getExtDate1();</code>
Parameters	None.
Discussion	

setExtDate1()

Specifies the first extension date.

Syntax	<code>public void setExtDate1(Date <i>date</i>);</code>
Parameters	<i>date</i> — A date that indicates the first extension date.
Discussion	

getExtDate2()

Retrieves the second extension date.

Syntax	<code>public Date getExtDate2();</code>
Parameters	None.
Discussion	

setExtDate2()

Specifies the second extension date.

Syntax	<code>public void setExtDate2 (Date <i>date</i>);</code>
Parameters	<i>date</i> — A date that indicates the second extension date.
Discussion	

getExtDate3()

Retrieves the third extension date.

Syntax	<code>public Date getExtDate3();</code>
Parameters	None.
Discussion	

setExtDate3()

Specifies the third extension date.

Syntax	<code>public void setExtDate3 (Date <i>date</i>);</code>
Parameters	<i>date</i> — A date that indicates the third extension date.
Discussion	

getExtDate4()

Retrieves the fourth extension date.

Syntax	<code>public Date getExtDate4();</code>

Parameters	None.

Discussion

setExtDate4()

Specifies the fourth extension date.

Syntax	<code>public void setExtDate4 (Date <i>date</i>);</code>

Parameters	<i>date</i> — A date that indicates the fourth extension date.

Discussion

getExtNum1()

Retrieves the first extension number.

Syntax	<code>public double getExtNum1();</code>

Parameters	None.

Discussion

setExtNum1()

Specifies the first extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>

Parameters	<i>num</i> — A double that indicates the first extension number.

Discussion

getExtNum2()

Retrieves the second extension number.

Syntax	<code>public double getExtNum2();</code>

Parameters	None.

Discussion

setExtNum2()

Specifies the second extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the second extension number.
Discussion	

getExtNum3()

Retrieves the third extension number.

Syntax	<code>public double getExtNum3();</code>
Parameters	None.
Discussion	

setExtNum3()

Specifies the third extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>
Parameters	<i>num</i> — A double that indicates the third extension number.
Discussion	

getExtNum4()

Retrieves the fourth extension number.

Syntax	<code>public double getExtNum4();</code>
Parameters	None.
Discussion	

setExtNum4()

Specifies the fourth extension number.

Syntax	<code>public void setExtNum1 (double <i>num</i>);</code>

Parameters

num — A double that indicates the fourth extension number.

Discussion

Membership Component Business Object Interfaces

This chapter discusses component business object interface classes for membership records. These classes provide access to records stored on a Directory Server database using the Lightweight Directory Access Protocol (LDAP).

This chapter includes the following sections:

- Overview
- Membership Class Reference Descriptions

Overview

Membership classes represent LDAP records on the Directory Server. You typically call a class method of the `BXMembershipMgr` class to create an object and call the object's `setMembershipContext()` method to associate it with a context.

Once the object has been created and associated with a context, you can use the Returns access methods described in “IBXEntity Class Reference” to create, retrieve, update, and delete a record in the Directory Server database.

Membership Classes

Membership classes represent LDAP records on the Directory Server. You use these classes to manipulate membership objects. Table 3-1 lists these classes.

Table 3-1 Membership Classes/Interfaces

Class Name	Description	Location
1. BXAddress	Implementation of IBXAddress.	page 306
2. IBXAddress	Interface that defines address objects.	page 307
3. BXBillerProfile	Implementation of IBXBillerProfile.	page 312
4. IBXBillerProfile	Interface that defines billing organization objects.	page 312
5. BXCumulativeProfile	Implementation of IBXCumulativeProfile.	page 336
6. IBXCumulativeProfile	Interface that defines customer profile objects.	page 337
7. BXFinancialAccount	Implementation of IBXFinancialAccount.	page 362
8. IBXFinancialAccount	Manipulates financial account objects.	page 362
9. BXMembershipContext	Establishes a context between an AppLogic and a membership business object.	page 367
10. BXMembershipEntity	Base class that is subclassed by all membership classes.	page 371
11. IBXMembershipEntity	Base interface for membership business objects.	page 372
12. BXMembershipMgr	Creates membership-related business objects.	page 373
13. BXPaymentType	Implementation of IBXPaymentType.	page 377
14. IBXPaymentType	Interface that defines payment type objects.	page 377

BXAddress Class Reference

The BXAddress class is an implementation of IBXAddress interface, which defines address objects.

Whenever possible, use the interface class IBXAddress (see “IBXAddress Class Reference”) instead of BXAddress.

IBXAddress Class Reference

The IBXAddress interface defines address objects that represent customer address record objects. You call the `BXMembershipMgr.createAddress()` method to create an address object.

Class Definition

Table 3-2 describes the IBXAddress interface definition.

Table 3-2 IBXAddress Interface Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.membership;</code>
Syntax	<code>public interface IBXAddress extends IBXMembershipEntity {...}</code>

Methods

This section describes the IBXAddress methods. Table 3-3 lists these methods.

Table 3-3 IBXAddress Class Methods

Method	Description
Getting and setting address information	
<code>getType()</code>	Determines the address type.
<code>setType()</code>	Specifies the address type.
<code>getAddress1()</code>	Determines the first line of the address.
<code>setAddress1()</code>	Specifies the first line of the address.
<code>getAddress2()</code>	Determines the second line of the address.
<code>setAddress2()</code>	Specifies the second line of the address.
<code>getAddress3()</code>	Determines the third line of the address.
<code>setAddress3()</code>	Specifies the third line of the address.
<code>getCity()</code>	Determines the city.
<code>setCity()</code>	Specifies the city.
<code>getState()</code>	Determines the state.
<code>setState()</code>	Specifies the state.
<code>getPostalCode()</code>	Determines the zip or postal code.

Table 3-3 IBXAddress Class Methods

Method	Description
setPostalCode ()	Specifies the zip or postal code.
getCountry ()	Determines the country.
setCountry ()	Specifies the country.
getAddressStatus ()	Determines the address status.
setAddressStatus ()	Specifies the address status.

getAddress1()

Determines the first line of the address.

Syntax	<code>String getAddress () ;</code>
Parameters	None.
Discussion	Returns a string that contains the first line of the address.

getAddress2()

Determines the second line of the address.

Syntax	<code>String getAddress2 () ;</code>
Parameters	None.
Discussion	Returns a string that contains the second line of the address.

getAddress3()

Determines the third line of the address.

Syntax	<code>String getAddress3 () ;</code>
Parameters	None.
Discussion	Returns a string that contains the third line of the address.

getAddressStatus()

Determines the address status.

Syntax	<code>String getAddressStatus();</code>
Parameters	None.
Discussion	Returns a string that contains the address status.

getCity()

Determines the city.

Syntax	<code>String getCity();</code>
Parameters	None.
Discussion	Returns a string that contains the city of the address.

getCountry()

Determines the country.

Syntax	<code>String getCountry();</code>
Parameters	None.
Discussion	Returns a string that specifies the country of the address.

getPostalCode()

Determines the zip or postal code.

Syntax	<code>String getPostalCode();</code>
Parameters	None.
Discussion	Returns a string that specifies the zip code of the address.

getState()

Determines the state.

Syntax	<code>String getState();</code>
Parameters	None.
Discussion	Returns a string that specifies the state of the address.

getType()

Determines the address type.

Syntax	<code>String getType();</code>
Parameters	None.
Discussion	Returns a string that specifies the type of the address.

setAddress1()

Specifies the first line of the address.

Syntax	<code>void setAddress1 (String value);</code>
Parameters	<i>value</i> — A string that specifies the first line of the address.
Discussion	

setAddress2()

Specifies the second line of the address.

Syntax	<code>void setAddress2 (String value);</code>
Parameters	<i>value</i> — A string that specifies the second line of the address.
Discussion	

setAddress3()

Specifies the third line of the address.

Syntax	<code>void setAddress3 (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the third line of the address.

setAddressStatus()

Specifies the address status.

Syntax	<code>void setAddressStatus (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the address status.
Discussion	

setCity()

Specifies the city.

Syntax	<code>void setCity (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the city
Discussion	

setCountry()

Specifies the country.

Syntax	<code>void setCountry (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the country
Discussion	

setPostalCode()

Specifies the postal code.

Syntax	<code>void setPostalCode (String value) ;</code>

Parameters	<i>value</i> — A string that specifies the postal code.
Discussion	

setState()
Specifies the state.

Syntax	<code>void setState (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the state.
Discussion	

setType()
Specifies the address type.

Syntax	<code>void setType (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the address type.
Discussion	

BXBillerProfile Class Reference

The BXBillerProfile class is an implementation of IBXBillerProfile interface, which defines billing organization objects.

Whenever possible, use the interface class IBXBillerProfile (see “IBXBillerProfile Class Reference”) instead of BXBillerProfile.

IBXBillerProfile Class Reference

The IBXBillerProfile interface defines billing organization objects that represent biller profile records. You call the `BXMembershipMgr.createBillerProfile()` method to create a biller profile object.

Before you can call the `IBXEntity.ejbCreate()` method (see page 129) to create a new record Returns associated with the object, you must call the `setOrgName()` method to set the object’s account type field.

Interface Definition

Table 3-4 describes the IBXBillerProfile interface definition.

Table 3-4 IBXBillerProfile Interface Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.membership;</code>
Syntax	<code>public interface IBXBillerProfile extends IBXMembershipEntity {...}</code>

Methods

This section describes the IBXBillerProfile class method. Table 3-5 lists these methods.

Table 3-5 IBXBillerProfile Class Methods (1 of 5)

Methods	Description
Retrieving biller-related records	
<code>ejbFindByOrganizationName()</code>	Retrieves the biller profile record specified by the organization name.
<code>ejbFindAllPersons()</code>	Retrieves the customer profile records associated with this biller.
<code>ejbFindAllCompanies()</code>	Retrieves the companies belonging to this organization.
<code>ejbFindAllFinancialAccounts()</code>	Retrieves the financial account records associated with this biller.
<code>ejbFindAllPaymentTypes()</code>	Retrieves the payment type records associated with this biller.
<code>ejbFindAllAccountTypes()</code>	Retrieves the account type records associated with this biller.
<code>ejbCreateAccountType()</code>	Creates an account type record for this biller in the database.
<code>ejbRemoveAccountType()</code>	Removes an account type record for this biller from the database.
<code>ejbStoreAccountType()</code>	Updates an account type record for this biller in the database.
<code>ejbCreateFinancialAccount()</code>	Creates a financial account record for this biller in the database.
<code>ejbRemoveFinancialAccount()</code>	Removes a financial account record for this biller from the database.
<code>ejbStoreFinancialAccount()</code>	Updates a financial account record for this biller in the database.

Table 3-5 IBXBillerProfile Class Methods (2 of 5)

Methods	Description
<code>ejbCreatePaymentType()</code>	Creates a payment type record for this biller in the database.
<code>ejbRemovePaymentType()</code>	Removes a payment type record for this biller from the database.
<code>ejbStorePaymentType()</code>	Updates a payment type record for this biller in the database.
<code>ejbFindByOrganizationName</code>	Retrieves the organization name
<code>ejbFindAllBillers</code>	Retrieves all the billers
<code>ejbLoadAccountType</code>	Loads the biller's account types
<code>ejbLoadPaymentType</code>	Retrieves the biller's payment types
Getting and setting biller Profile information	
<code>getIG()</code>	Determines the IG user
<code>setIG()</code>	Specifies the IG user
<code>getIGURL()</code>	Determines the IG URL destination
<code>setIGURL()</code>	Sets the URL for the IG Server
<code>getType()</code>	Determines the kind of biller.
<code>setType()</code>	Specifies the kind of biller.
<code>getTabledefFilepath()</code>	Determines the path to the definition table
<code>setTabledefFilepath()</code>	Sets the path to the definition table
<code>getPrimaryContactPhone()</code>	Determines the primary phone #
<code>setPrimaryContactPhone()</code>	Sets the primary phone #
<code>getPrimaryContactEmail()</code>	Determines the primary email
<code>setPrimaryContactEmail()</code>	Sets the primary email
<code>getSecondaryContactPhone()</code>	Determines the secondary phone
<code>setSecondaryContactPhone()</code>	Sets the secondary phone
<code>getSecondaryContactEmail()</code>	Determines the secondary email
<code>setSecondaryContactEmail()</code>	Sets the secondary email
<code>getEnrollOption()</code>	Determines the enrollment option
<code>setEnrollOption()</code>	Sets the enrollment option
<code>getBillPresentment()</code>	Determines which bills to present
<code>setBillPresentment()</code>	Sets what bills to present

Table 3-5 IBXBillerProfile Class Methods (3 of 5)

Methods	Description
<code>getAutopay()</code>	Determines the autopay option
<code>setAutopay()</code>	Sets the autopay option
<code>getOrgName()</code>	Determines the biller's organization name.
<code>setOrgName()</code>	Specifies the biller's organization name.
<code>getBusinessCategory()</code>	Determines the biller's business category.
<code>setBusinessCategory()</code>	Specifies the biller's business category.
<code>getDescription()</code>	Determines the biller's description.
<code>setDescription()</code>	Specifies the biller's description.
<code>getAddress()</code>	Determines the biller's address.
<code>setAddress()</code>	Specifies the biller's address.
<code>getPhone()</code>	Determines the biller's phone number.
<code>setPhone()</code>	Specifies the biller's phone number.
<code>getFaxNumber()</code>	Determines the biller's fax number.
<code>setFaxNumber()</code>	Specifies the biller's fax number.
<code>getEmail()</code>	Determines the biller's email address.
<code>setEmail()</code>	Specifies the biller's email address.
<code>getURL()</code>	Determines the biller's URL.
<code>setURL()</code>	Specifies the biller's URL.
<code>getLanguage()</code>	Determines the biller's language.
<code>setLanguage()</code>	Specifies the biller's language.
<code>getCurrency()</code>	Determines the biller's currency.
<code>setCurrency()</code>	Specifies the biller's currency.
<code>getCharSet()</code>	Determines the biller's character set.
<code>setCharSet()</code>	Specifies the biller's character set.
<code>getTimeZone()</code>	Determines the biller's time zone.
<code>setTimeZone()</code>	Specifies the biller's time zone.
<code>getBillerStatus()</code>	Determines the biller's status.
<code>setBillerStatus()</code>	Specifies the biller's status.
<code>getCreateDate()</code>	Determines the biller profile's creation date.

Table 3-5 IBXBillerProfile Class Methods (4 of 5)

Methods	Description
<code>setCreateDate()</code>	Sets the biller profile's creation date
<code>getModifyDate()</code>	Determines the biller profile's modification date.
<code>setModifyDate()</code>	Set the biller profile's modification date
<code>getPrimaryContact()</code>	Determines the biller's primary contact.
<code>setPrimaryContact()</code>	Specifies the biller's primary contact.
<code>getSecondaryContact()</code>	Determines the biller's secondary contacts.
<code>setSecondaryContact()</code>	Specifies the biller's secondary contacts.
<code>getProfileExtension()</code>	Determines the biller's profile extension.
<code>setProfileExtension()</code>	Specifies the biller's profile extension.
<code>getServiceExtension()</code>	Determines the biller's service extension.
<code>setServiceExtension()</code>	Specifies the biller's service extension.
<code>getCreatedBy()</code>	Determines the biller profile's creator.
<code>setCreatedBy()</code>	Specifies the biller profile's creator.
<code>getTaxID()</code>	Determines the biller's tax ID.
<code>setTaxID()</code>	Specifies the biller's tax ID.
<code>getExtension()</code>	Determines the biller's extension.
<code>setExtension()</code>	Specifies the biller's extension.
<code>getMerchantID()</code>	Determines the biller's merchant ID.
<code>setMerchantID()</code>	Specifies the biller's merchant ID.
<code>getPayeeID()</code>	Determines the payee's id
<code>setPayeeID()</code>	Sets the payee's id
Multiple Biller Extensions	
<code>getIndustryProfile()</code>	Determines the Industry profile of the billers
<code>setIndustryProfile()</code>	Sets the Industry profile of the billers
<code>getDatabase()</code>	Determines the biller database
<code>setDatabase()</code>	Sets the biller database
<code>getUserID()</code>	Determines the user id of the biller
<code>setUserID()</code>	Sets the user id of the biller
<code>getPassword()</code>	Determines the password of the biller

Table 3-5 IBXBillerProfile Class Methods (5 of 5)

Methods	Description
<code>setPassword()</code>	Sets the password of the biller
<code>getTemplatePath()</code>	Determines the path for a specific biller
<code>setTemplatePath()</code>	Sets the path for a specific biller
<code>getCycleDate()</code>	Determines the cycle date for billers
<code>setCycleDate()</code>	Sets the cycle date for billers
<code>getMonthHistory()</code>	Determines the history date
<code>setMonthHistory()</code>	Sets the history date
<code>getBillType()</code>	Determines the bill type for the billers
<code>setBillType()</code>	Sets the bill type for the billers
<code>getGroup()</code>	Determines the group of the billers
<code>SetGroup()</code>	Sets the billers group
<code>getMaxRows()</code>	Determines the maximum rows for the billers
<code>setMaxRows()</code>	Sets the maximum rows for the billers

ejbCreateAccountType()

Creates an account type record for this biller in the database.

Syntax	<code>String ejbCreateAccountType (IBXAccountType <i>accountType</i>)</code> <code>throws Exception;</code>
Parameters	<i>accountType</i> — The IBXAccountType object to store in the database.
Discussion	Both the biller profile object's database context and the account type's database context must previously have been set by calling each object's <code>createMembershipContext()</code> method. The organization name must be set in the biller profile object and the type field must be set in the account type object. Returns a string that contains the object ID for the newly-created account type record.

ejbCreateFinancialAccount()

Creates a financial account record for this biller in the database.

Syntax	<code>String ejbCreateFinancialAccount (IBXFinancialAccount <i>account</i>) throws Exception;</code>
Parameters	<i>account</i> — The IBXFinancial Account object to store in the database.
Discussion	Both the biller profile object's database context and the account type's database context must previously have been set by calling each object's <code>createMembershipContext()</code> method. The organization name must be set in the biller profile object and the type field must be set in the financial type object. Returns a string that contains the object ID for the newly-created financial account record.

ejbCreatePaymentType()

Creates a payment type record for this biller in the database.

Syntax	<code>String ejbCreatePaymentType (IBXPaymentType <i>paymentType</i>) throws Exception;</code>
Parameters	<i>paymentType</i> — The IBXPaymentType object to store in the database.
Discussion	Both the biller profile object's database context and the account type's database context must previously have been set by calling each object's <code>createMembershipContext()</code> method. The organization name must be set in the biller profile object and the type field must be set in the payment type object. Returns a string that contains the object ID for the newly-created account type record.

ejbFindAllCompanies()

Retrieves the companies belonging to this organization.

Syntax	<code>public Vector ejbFindAllCompanies (BXFilter <i>filter</i>, BXOrder <i>order</i>) throws Exception;</code>
Parameters	<i>filter</i> — The BXfilter object that specifies the Where clause in the SQL Select Statement. <i>order</i> — A BXOrder object that specifies the Order By clause of an SQL Select Statement

Discussion	The <code>ejbFindAllCompanies()</code> method retrieves all records for the companies belonging to this organization that match the clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, the <code>ejbFindAllCompanies()</code> method creates an object, sets the fields in the object from the record, and adds the object to the return vector. If the <i>filter</i> parameter is null, all records will be retrieved. If the <i>order</i> parameter is null, the retrieved records will be sorted in ascending order by their primary keys. The customer profile object's database contact must previously been set by calling the customer profile object's <code>createMembershipContact()</code> method. Returns a vector that contains the object <code>BXCustomerProfile</code> objects representing selected records.

ejbFindAllAccountTypes()

Retrieves the account type records associated with this biller.

Syntax	<code>Vector ejbFindAllAccountTypes(BXFilter <i>filter</i>, BXOrder <i>order</i>) throws Exception;</code>
Parameters	<i>filter</i> — The <code>BXFilter</code> object that specifies the Where clause of an SQL Select statement. <i>order</i> — A <code>BXOrder</code> object that specifies the Order By clause of an SQL Select statement
Discussion	The <code>ejbFindAllAccountTypes()</code> method retrieves all records for the biller that match the clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, the <code>ejbFindAllAccountTypes()</code> method creates an object, sets the fields in the object from the record, and adds the object to the return vector. If the <i>filter</i> parameter is null, all records will be retrieved. If the <i>order</i> parameter is null, the retrieved records will be sorted in ascending order by their primary key. The biller profile object's database context must previously have been set by calling the biller profile object's <code>createMembershipContext()</code> method. Returns a vector that contains the object <code>BXAccountType</code> objects representing the selected records.

ejbFindAllFinancialAccounts()

Retrieves the financial account records associated with this biller.

Syntax	<code>Vector ejbFindAllFinancialAccounts(BXFilter <i>filter</i>, BXOrder <i>order</i>) throws Exception;</code>

Parameters	<i>filter</i> — The BXFilter object that specifies the Where clause of an SQL Select statement. <i>order</i> — A BXOrder object that specifies the Order By clause of an SQL Select statement
Discussion	The <code>ejbFindAllFinancialAccounts()</code> method retrieves all records for the biller that match the clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, the <code>ejbFindAllFinancialAccounts()</code> method creates an object, sets the fields in the object from the record, and adds the object to the return vector. If the <i>filter</i> parameter is null, all records will be retrieved. If the <i>order</i> parameter is null, the retrieved records will be sorted in ascending order by their primary key. The biller profile object's database context must previously have been set by calling the biller profile object's <code>createMembershipContext()</code> method. Returns a vector that contains the object BXAccountType objects representing the selected records.

ejbFindAllPaymentTypes()

Retrieves the payment type records associated with this biller.

Syntax	<code>Vector ejbFindAllPaymentTypes(BXFilter <i>filter</i>, BXOrder <i>order</i>)</code> <code>throws Exception;</code>
Parameters	<i>filter</i> — The BXFilter object that specifies the Where clause of an SQL Select statement. <i>order</i> — A BXOrder object that specifies the Order By clause of an SQL Select statement.
Discussion	The <code>ejbFindAllPaymentTypes()</code> method retrieves all records for the biller that match the clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, the <code>ejbFindAllPaymentTypes()</code> method creates an object, sets the fields in the object from the record, and adds the object to the return vector. If the <i>filter</i> parameter is null, all records will be retrieved. If the <i>order</i> parameter is null, the retrieved records will be sorted in ascending order by their primary key. The biller profile object's database context must previously have been set by calling the biller profile object's <code>createMembershipContext()</code> method. Returns a vector that contains the object BXPaymentType objects representing the selected records.

ejbFindAllPersons()

Retrieves the customer profile records associated with this biller.

Syntax	<code>Vector ejbFindAllPersons (BXFilter <i>filter</i>, BXOrder <i>order</i>)</code> throws <code>Exception</code> ;
Parameters	<i>filter</i> — The BXFilter object that specifies the Where clause of an SQL Select statement. <i>order</i> — A BXOrder object that specifies the Order By clause of an SQL Select statement.
Discussion	The <code>ejbFindAllPersons()</code> method retrieves all records for the biller that match the clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, the <code>ejbFindAllPersons()</code> method creates an object, sets the fields in the object from the record, and adds the object to the return vector. If the <i>filter</i> parameter is null, all records will be retrieved. If the <i>order</i> parameter is null, the retrieved records will be sorted in ascending order by their primary key. The biller profile object's database context must previously have been set by calling the biller profile object's <code>createMembershipContext()</code> method. Returns a vector that contains the object <code>BXCustomerProfile</code> objects representing the selected records.

ejbFindByOrganizationName()

Retrieves the biller profile record specified by the organization name.

Syntax	<code>void ejbFindByOrganizationName (String <i>name</i>)</code> throws <code>Exception</code> ;
Parameters	<i>name</i> — A string that contains the name of the biller.
Discussion	The <i>name</i> parameter specifies the record to read. The fields of this object are set from the contents of the record. The biller profile object's database context must previously have been set by calling the biller profile object's <code>createMembershipContext()</code> method. This method is the same as the <code>ejbFindByPrimaryKey()</code> method.

Warning	Previous contents of objects are replaced by contents of the selected record.

ejbRemoveAccountType()

Removes an account type record for this biller from the database.

Syntax	<code>void ejbRemoveAccountType(IBXAccountType <i>accountType</i>) throws Exception;</code>
Parameters	<i>accountType</i> — The IBXAccountType object to remove from the database
Discussion	The fields of the object itself are not changed. Both the biller profile object's database context and the account type object's database context must previously have been set by calling each object's <code>createMembershipContext ()</code> method.

ejbRemoveFinancialAccount()

Removes a financial account record for this biller from the database.

Syntax	<code>void ejbRemoveFinancialAccount(IBXFinancialAccount <i>Returns account</i>) throws Exception;</code>
Parameters	<i>Returns account</i> — The IBXFinancialAccount object to remove from the database
Discussion	The fields of the object itself are not changed. Both the biller profile object's database context and the financial account object's database context must previously have been set by calling each object's <code>createMembershipContext ()</code> method.

ejbRemovePaymentType()

Removes a payment type record for this biller from the database.

Syntax	<code>void ejbRemovePaymentType(IBXPaymentType <i>paymentType</i>) throws Exception;</code>
Parameters	<i>paymentType</i> — The IBXPaymentType object to remove from the database.
Discussion	The fields of the object itself are not changed. Both the biller profile object's database context and the financial account object's database context must previously have been set by calling each object's <code>createMembershipContext ()</code> method.

ejbStoreAccountType()

Updates an account type record for this biller in the database.

Syntax	<code>void ejbStoreAccountType(IBXAccountType <i>Returns accountType</i>) throws Exception;</code>
Parameters	<i>Returns accountType</i> — The IBXFinancialAccountType to update from the database
Discussion	Both the biller profile object's database context and the account type object's database context must previously have been set by calling each object's <code>createMembershipContext()</code> method.

Warning	You must retrieve the record before you update it; otherwise, fields that you do not explicitly set will be written as null.

ejbStoreFinancialAccount()

Updates a financial account record for this biller in the database.

Syntax	<code>void ejbStoreFinancialAccountType(IBXFinancialAccount <i>Returns account</i>) throws Exception;</code>
Parameters	<i>Returns account</i> — The IBXFinancialAccount object to update from the database.
Discussion	Both the biller profile object's database context and the account type object's database context must previously have been set by calling each object's <code>createMembershipContext()</code> method.

Warning	You must retrieve the record before you update it; otherwise, fields that you do not explicitly set will be written as null.

ejbStorePaymentType()

Updates a payment type record for this biller in the database.

Syntax	<code>void ejbStorePaymentType(IBXPaymentType <i>paymentType</i>) throws Exception;</code>
Parameters	<i>paymentType</i> — The IBXPaymentType to update from the database

Discussion	Both the biller profile object's database context and the account type object's database context must previously have been set by calling each object's <code>createMembershipContext()</code> method.

Warning	You must retrieve the record before you update it; otherwise, fields that you do not explicitly set will be written as null.

ejbFindByOrganizationName

Retrieves billers by organization name.

Syntax	<code>public void ejbFindByOrganizationName(String <i>name</i>) throws Exception;</code>
Parameters	<i>name</i> — A string that contains the name of the biller.
Discussion	The <i>name</i> parameter specifies the record to read. The fields of this object are set from the contents of the record. The biller profile object's database context must previously have been set by calling the biller profile object's <code>createMembershipContext()</code> method. This method is the same as the <code>ejbFindByPrimaryKey()</code> method.

ejbFindAllBillers

Retrieves all the billers in the system.

Syntax	<code>public Vector ejbFindAllBillers(BxFilter <i>filter1</i>, BxFilter <i>filter2</i>, BxOrder <i>order</i>, String <i>startIndex</i>, int <i>maxRows</i>) throws Exception;</code>
Parameters	<i>filter1</i> — This is filter1 parameter <i>filter2</i> — This filter2 parameter <i>order</i> — This is the order of biller discovery <i>startIndex</i> — The beginning of the find <i>maxRows</i> — The number of rows in the find.
Discussion	

ejbLoadAccountType

Loads the Biller's account types.

Syntax	<code>public void ejbLoadAccountType(IBXAccountType <i>Returns accountType</i>) throws Exception;</code>
Parameters	<i>Returns accountType</i> — The IBXAccountType to update in the database
Discussion	Both the biller profile object's database context and the account type object's database context must previously have been set by calling each object's <code>createMembershipContext()</code> method.

Warning	You must retrieve the record before you update it; otherwise, fields that you do not explicitly set will be written as null.

ejbLoadPaymentType

Retrieves the biller's payment types.

Syntax	<code>public void ejbLoadPaymentType(IBXPaymentType <i>paymentType</i>) throws Exception;</code>
Parameters	<i>paymentType</i> — The IBXPaymentType object to store in the database
Discussion	Both the biller profile object's database context and the account type object's database context must previously have been set by calling each object's <code>createMembershipContext()</code> method.

getAddress()

Determines the biller's address.

Syntax	<code>IBXAddress getAddress();</code>
Parameters	None.
Discussion	Returns an IBXAddress object that contains the address.

getBillerStatus()

Determines the biller's status.

Syntax	<code>String getBillerStatus();</code>

Parameters	None.
Discussion	Returns a string that contains the status.

getBusinessCategory()
Determines the biller’s business category.

Syntax	<code>String getBusinessCategory();</code>
Parameters	None.
Discussion	Returns a string that contains the business category.

getCharSet()
Determines the biller’s character set.

Syntax	<code>String getCharSet();</code>
Parameters	None.
Discussion	Returns a string that contains the character set.

getCreateDate()
Determines the biller profile’s creation date.

Syntax	<code>Date getCreateDate();</code>
Parameters	None.
Discussion	Returns a string that contains the creation date.

getCreatedBy()
Determines the biller profile’s creator.

Syntax	<code>String getCreatedBy();</code>
Parameters	None.
Discussion	Returns a string that contains the user name of the creator.

getCurrency()

Determines the biller's currency.

Syntax	<code>String getCurrency();</code>
Parameters	None.
Discussion	Returns a string that contains the currency.

getDescription()

Determines the biller's description.

Syntax	<code>String getDescription();</code>
Parameters	None.
Discussion	Returns a string that contains the description.

getEmail()

Determines the biller's email address.

Syntax	<code>String getEmail();</code>
Parameters	None.
Discussion	Returns a string that contains the email address.

getExtension()

Determines the biller's extension.

Syntax	<code>String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the extension

getFaxNumber()

Determines the biller’s fax number.

Syntax	<code>String getFaxNumber() ;</code>
Parameters	None.
Discussion	Returns a string that contains the fax number

getLanguage()

Determines the biller’s language.

Syntax	<code>String getLanguage() ;</code>
Parameters	None.
Discussion	Returns a string that contains the language.

getMerchantID()

Determines the biller’s merchant ID.

Syntax	<code>String getMerchantID() ;</code>
Parameters	None.
Discussion	Returns a string that contains the merchant ID.

getModifyDate()

Determines the biller profile’s modification date.

Syntax	<code>String getModifyDate() ;</code>
Parameters	None.
Discussion	Returns a string that contains the modification date.

getOrgName()

Determines the biller's organization name.

Syntax	<code>String getOrgName();</code>
Parameters	None.
Discussion	Returns a string that contains the organization name.

getPhone()

Determines the biller's phone number.

Syntax	<code>String getPhone();</code>
Parameters	None.
Discussion	Returns a string that contains the biller's phone number.

getPrimaryContact()

Determines the biller's primary contact.

Syntax	<code>String getPrimaryContact();</code>
Parameters	None.
Discussion	Returns a string that contains the primary contact.

getProfileExtension()

Determines the biller's profile extension.

Syntax	<code>String getProfileExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the profile extension.

getSecondaryContact()

Determines the biller’s secondary contacts.

Syntax	<code>Vector getSecondaryContact () ;</code>
Parameters	None.
Discussion	Returns a vector that contains the secondary contact.

getServiceExtension()

Determines the biller’s service extension.

Syntax	<code>String getServiceExtension () ;</code>
Parameters	None.
Discussion	Returns a string that contains the service extension.

getTaxID()

Determines the biller’s tax ID.

Syntax	<code>String getTaxID () ;</code>
Parameters	None.
Discussion	Returns a string that contains the tax ID.

getTimeZone()

Determines the biller’s time zone.

Syntax	<code>String getTimeZone () ;</code>
Discussion	Returns a string that contains the time zone.

getType()

Determines the kind of biller.

Syntax	<code>String getType () ;</code>

Parameters	None.
Discussion	Returns a string that contains the biller type.

getURL()

Determines the biller's URL.

Syntax	<code>String getURL();</code>
Parameters	None.
Discussion	Returns a string that contains the biller's URL.

setAddress()

Specifies the biller's address .

Syntax	<code>void setAddress(IBXAddress value);</code>
Parameters	<i>value</i> — An IBXAddress object that specifies the address
Discussion	This method only changes the value of the address field in the object; it does not modify the value in the database.

setBillerStatus()

Specifies the biller's status.

Syntax	<code>void setBillerStatus(String value);</code>
Parameters	<i>value</i> — An object that specifies the status
Discussion	This method only changes the value of the status field in the object; it does not modify the value in the database.

setBusinessCategory()

Specifies the biller's business category.

Syntax	<code>void setBusinesscategory (String value);</code>
Parameters	<i>value</i> — A string that specifies the category

Discussion	This method only changes the value of the address field in the object; it does not modify the value in the database.

setCharSet()

Specifies the biller’s character set.

Syntax	<code>void setCharSet (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the character set
Discussion	This method only changes the value of the character set field in the object; it does not modify the value in the database.

setCreatedBy()

Specifies the biller profile’s creator.

Syntax	<code>void setCreatedBy (String value) ;</code>
Parameters	<i>value</i> — A object that specifies the creator
Discussion	This method only changes the value of the address field in the object; it does not modify the value in the database.

setCurrency()

Specifies the biller’s currency.

Syntax	<code>void setCurrency(String value) ;</code>
Parameters	<i>value</i> — A string that specifies the currencye
Discussion	This method only changes the value of the currency field in the object; it does not modify the value in the database.

setDescription()

Specifies the biller’s description.

Syntax	<code>void setDescription (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the description

Discussion	This method only changes the value of the description field in the object; it does not modify the value in the database.

setEmail()

Specifies the biller's email address.

Syntax	<code>void setEmail (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the email address
Discussion	This method only changes the value of the email field in the object; it does not modify the value in the database.

setExtension()

Specifies the biller's extension.

Syntax	<code>void setExtension (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the email address
Discussion	This method only changes the value of the extension field in the object; it does not modify the value in the database.

setFaxNumber()

Specifies the biller's fax number.

Syntax	<code>void setFaxNumber (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the fax number
Discussion	This method only changes the value of the fax number field in the object; it does not modify the value in the database.

setLanguage()

Specifies the biller's language.

Syntax	<code>void setLanguage (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the language

Discussion	This method only changes the value of the language field in the object; it does not modify the value in the database.

setMerchantID()

Specifies the biller’s merchant ID.

Syntax	<code>void setMerchantID (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the merchant ID
Discussion	This method only changes the value of the ID field in the object; it does not modify the value in the database.

setOrgName()

Specifies the biller’s organization name.

Syntax	<code>void setOrgName (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the organization name
Discussion	This method only changes the value of the organization name field in the object; it does not modify the value in the database.

setPhone()

Specifies the biller’s phone number.

Syntax	<code>void setPhone (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the phone number.
Discussion	This method only changes the value of the phone number field in the object; it does not modify the value in the database.

setPrimaryContact()

Specifies the biller’s primary contact.

Syntax	<code>void setPrimaryContact (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the contact

Discussion	This method only changes the value of the contact field in the object; it does not modify the value in the database.

setProfileExtension()

Specifies the biller's profile extension.

Syntax	<code>void setProfileExtension (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the extension
Discussion	This method only changes the value of the extension field in the object; it does not modify the value in the database.

setSecondaryContact()

Specifies the biller's secondary contacts.

Syntax	<code>void setSecondaryContact (Vector <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the secondary contact
Discussion	This method only changes the value of the contact field in the object; it does not modify the value in the database.

setServiceExtension()

Specifies the biller's service extension.

Syntax	<code>void setServiceExtension (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the service extension
Discussion	This method only changes the value of the extension field in the object; it does not modify the value in the database.

setTaxID()

Specifies the biller's tax ID.

Syntax	<code>void setTaxID (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the tax ID.

Discussion	This method only changes the value of the tax ID field in the object; it does not modify the value in the database.

setTimeZone()

Determines the biller’s time zone.

Syntax	<code>void setTimeZone (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the time zone
Discussion	This method only changes the value of the time zone field in the object; it does not modify the value in the database.

setType()

Specifies the kind of biller.

Syntax	<code>void setType (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the type
Discussion	This method only changes the value of the type field in the object; it does not modify the value in the database.

setURL()

Specifies the biller’s URL.

Syntax	<code>void setURL (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the URL
Discussion	This method only changes the value of the URL field in the object; it does not modify the value in the database.

BXCustomerProfile Class Reference

The BXCustomerProfile class is an implementation of the IBXCustomerProfile interface, which defines customer profile objects.

Whenever possible, use the interface class IBXCustomerProfile (see “IBXCustomerProfile Class Reference”) instead of BXCustomerProfile.

IBXCustomerProfile Class Reference

The IBXCustomerProfile interface defines customer profile objects that represent customer profile records. You call the `BXMembershipMgr.createBillerProfile()` method to create a biller profile object.

Before you can call the `IBXEntity.ejbCreate()` method to create a new record Returns associated with the object, you must call the `setLoginName()`, `setPassword()`, `setLastName()`, `setFirstName()`, and `setOrgName()` methods to set the login name, password, last name, first name, and organization name fields, respectively.

Interface Definition

Table 3-6 describes the IBXCustomerProfile interface definition.

Table 3-6 IBXCustomerProfile Interface Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.membership;</code>
Syntax	<code>interface IBXCustomerProfile extends IBXMembershipEntity {...}</code>

Methods

This section describes the IBXCustomerProfile methods. Table 3-7 lists these methods

Table 3-7 IBXCustomerProfile Interface Methods (1 of 4)

Method	Description
Manipulating customer-related records	
<code>ejbFindByLoginName()</code>	Retrieves the record with the specified login name.
<code>ejbFindAllFinancialAccounts()</code>	Retrieves financial account records associated with this customer.
<code>ejbFindAllByIdList()</code>	Retrieves customer profile records with the specified IDs.
<code>ejbCreateFinancialAccount()</code>	Creates a financial account record for this customer.

Table 3-7 IBXCustomerProfile Interface Methods (2 of 4)

Method	Description
<code>ejbRemoveFinancialAccount()</code>	Removes a financial account record associated with this customer.
<code>ejbLoadFinancialAccount()</code>	Loads a financial account record
<code>ejbStoreFinancialAccount()</code>	Updates a financial account record associated with this customer.
Returns authenticating and protecting customer records	
<code>Returns authenticate()</code>	Verifies a customer's login name and password.
<code>updatePassword()</code>	Changes the customer's password.
Getting and setting customer profile information	
<code>getBillerID()</code>	Determines the Biller ID.
<code>getAccount()</code>	Determines the customer's account number.
<code>setAccount()</code>	Specifies the customer's account number.
<code>getPrefix()</code>	Determines the customer's name prefix.
<code>setPrefix()</code>	Specifies the customer's name prefix.
<code>getSuffix()</code>	Determines the customer's name suffix.
<code>setSuffix()</code>	Specifies the customer's name suffix.
<code>getLastName()</code>	Determines the customer's surname.
<code>setLastName()</code>	Specifies the customer's surname.
<code>getFirstName()</code>	Determines the customer's given name.
<code>setFirstName()</code>	Specifies the customer's given name.
<code>getMiddleName()</code>	Determines the customer's middle name.
<code>setMiddleName()</code>	Specifies the customer's middle name.
<code>getNickName()</code>	Determines the customer's nick name.
<code>setNickName()</code>	Specifies the customer's nick name.
<code>getDescription()</code>	Determines the customer's description.
<code>setDescription()</code>	Specifies the customer's description.
<code>getAddress()</code>	Determines the customer's address.
<code>setAddress()</code>	Specifies the customer's address.
<code>getDayPhone()</code>	Determines the customer's daytime phone number.

Table 3-7 IBXCustomerProfile Interface Methods (3 of 4)

Method	Description
setDayPhone()	Specifies the customer's daytime phone number.
getNightPhone()	Determines the customer's evening phone number.
setNightPhone()	Specifies the customer's evening phone number.
getFaxNumber()	Determines the customer's fax number.
setFaxNumber()	Specifies the customer's fax number.
getEmail()	Determines the customer's email address.
setEmail()	Specifies the customer's email address.
getSecretWord()	Determines the customer's secret word.
setSecretWord()	Specifies the customer's secret word.
getLoginName()	Determines the customer's login name.
setLoginName()	Specifies the customer's login name.
getPassword()	Determines the customer's password.
setPassword()	Specifies the customer's password.
getPasswordHint()	Determines the customer's password hint.
setPasswordHint()	Specifies the customer's password hint.
getPasswordLastChanged()	Determines the date the customer's password was last changed.
setPasswordLastChanged()	Specifies the date the customer's password was last changed.
getLanguage()	Determines the customer's language.
setLanguage()	Specifies the customer's language.
getCurrency()	Determines the customer's currency.
setCurrency()	Specifies the customer's currency.
getCharSet()	Determines the customer's character set.
setCharSet()	Specifies the customer's character set.
getTimeZone()	Determines the customer's time zone.
setTimeZone()	Specifies the customer's time zone.
getCustomerStatus()	Determines the customer's status.
setCustomerStatus()	Specifies the customer's status.
getCreateDate()	Determines the customer profile's creation date.
getModifyDate()	Determines the customer profile's modification date.

Table 3-7 IBXCustomerProfile Interface Methods (4 of 4)

Method	Description
<code>getCreatedBy()</code>	Determines the customer profile's creator.
<code>setCreatedBy()</code>	Specifies the customer profile's creator.
<code>getPrimaryContact()</code>	Determines the customer's primary contact.
<code>setPrimaryContact()</code>	Specifies the customer's primary contact.
<code>getSecondaryContact()</code>	Determines the customer's secondary contacts.
<code>setSecondaryContact()</code>	Specifies the customer's secondary contacts.
<code>getProfileExtension()</code>	Determines the customer's profile extension.
<code>setProfileExtension()</code>	Specifies the customer's profile extension.
<code>getServiceExtension()</code>	Determines the customer's service extension.
<code>setServiceExtension()</code>	Specifies the customer's service extension.
<code>getOrgName()</code>	Determines the customer's organization name.
<code>setOrgName()</code>	Specifies the customer's organization name.
<code>getExtension()</code>	Determines the customer's extension.
<code>setExtension()</code>	Specifies the customer's extension.
<code>getGroup()</code>	Determines the customer's groups.
<code>setGroup()</code>	Specifies the customer's groups.
<code>getAccountType()</code>	Determines the customer's account types.
<code>setAccountType()</code>	Specifies the customer's account types.
<code>getDefaultPmtAcct()</code>	Determines the customer default payment option.
<code>setDefaultPmtAcct()</code>	Sets the customer default payment option.
<code>getFileID()</code>	Determines the file ID.
<code>setFileID()</code>	Specifies the file ID.
Generic Methods	
<code>setAttribute()</code>	Specifies the attribute.
<code>callCreateAdmin()</code>	Creates an admin for the customer profile.
<code>callStoreAdmin()</code>	Stores an admin for the customer profile.

Returns authenticate()

Verifies a customer's login name and password.

Syntax	<code>String authenticate (String <i>login</i>, String <i>password</i>) throws Exception;</code>
Parameters	<i>login</i> — A string that contains the login name. <i>password</i> — A string that contains the password
Discussion	The <code>authenticate()</code> method throws an exception if the login name does not match the password.

callCreateAdmin()

Creates an Admin for a customer profile.

Syntax	<code>void callCreateAdmin (BXStatus <i>status</i>) throws Exception;</code>
Parameters	<i>status</i> — A BXStatus object
Discussion	The <code>callCreateAdmin()</code> method throws an exception if an Admin is not created.

callStoreAdmin()

Stores an Admin for a customer profile.

Syntax	<code>void callStoreAdmin () throws Exception;</code>
Parameters	None.
Discussion	The <code>callStoreAdmin()</code> method throws an exception if an Admin is not stored.

ejbCreateFinancialAccount()

Creates a financial account record for this customer.

Syntax	<code>String ejbCreateFinancialAccount (IBXFinancialAccount <i>Returns account</i>) throws Exception;</code>
Parameters	<i>Returns account</i> — An IBXFinancialAccount object to store in the database

Discussion	The customer profile object's database context must previously have been set by calling the customer profile object's <code>createMembershipContext()</code> method. The organization name must be set in the customer profile object and the type, payment handle, and pan handle fields must be set in the financial account object. Returns a string that contains the object ID for the newly-created financial account record.

ejbFindAllByIDList()

Retrieves customer profile records with the specified IDs.

Syntax	<code>Vector ejbFindAllByIDList(Vector idList) throws Exception;</code>
Parameters	<i>idList</i> — A vector object that specifies the ID
Discussion	The <code>ejbFindAllByIDList()</code> method retrieves all records for the customers whose uid fields match an element of the <i>idList</i> parameter, in ascending order. For each record, the <code>ejbFindAllByIDList()</code> method creates an object, sets the fields in the object from the record, Returns and adds the object to the return vector. The customer profile object's database context must previously have been set by calling the customer profile object's <code>createMembershipContext()</code> method. Returns a vector that contains <code>BXCustomerProfile</code> objects representing the selected records.

ejbFindAllFinancialAccounts()

Retrieves customer profile records with the specified IDs.

Syntax	<code>Vector ejbFindAllFinancialAccounts(BXFilter filter, BXOrder order) throws Exception;</code>
Parameters	<i>filter</i> — A <code>BXFilter</code> object that specifies the Where clause of an SQL Select statement. <i>order</i> — A <code>BXOrder</code> object that species the Order By clause of an SQL Select statement.

Discussion	The <code>ejbFindAllFinancialAccounts()</code> method retrieves all records for the customer that match the clause in the <i>filter</i> parameter, ordered by the clause specified in the <i>order</i> parameter. For each record, the <code>ejbFindAllFinancialAccounts()</code> method creates an object, sets the fields in the object from the record, and adds the object to the return vector. If the <i>filter</i> parameter is null, all records will be retrieved. If the <i>order</i> parameter is null, the retrieved records will be sorted in Returns ascending order by their primary key. Both the customer profile object's database context and the financial Returns account object's database context must previously have been set by calling each object's <code>createMembershipContext()</code> method. Returns a vector that contains <code>BXFinancialAccount</code> objects representing the selected records.

ejbFindByLoginName()

Retrieves the record with the specified login name.

Syntax	<code>void ejbFindByLoginName (String <i>loginName</i>) throws Exception;</code>
Parameters	<i>loginName</i> — A string that contains the login name
Discussion	The <i>loginName</i> parameter specifies the record to read. The fields of this object are set from the contents of the record. The customer profile object's database context must previously have been set by calling the customer profile object's <code>createMembershipContext()</code> method. This method is the same as the <code>ejbFindByPrimaryKey()</code> method.

Warning	The previous contents of this object are replaced by the contents of the selected record.

ejbRemoveFinancialAccount()

Removes a financial account record associated with this customer.

Syntax	<code>void ejbRemoveFinancialAccount (IBXFinancialAccount <i>Returns_account</i>) throws Exception;</code>
Parameters	<i>Returns_account</i> — The <code>IBXFinancialAccount</code> object to remove from the database

Discussion	The fields of the object itself are not changed. Both the customer profile object's database context and the financial account object's database context must previously have been set by calling each object's <code>createMembershipContext ()</code> method.

ejbLoadFinancialAccount()

Loads a financial account.

Syntax	<code>void ejbLoadFinancialAccount (IBXFinancialAccount Returns_account) throws Exception;</code>
Parameters	<i>Returns_account</i> — The IBXFinancialAccount object to load from the database
Discussion	The fields of the object itself are not changed. Both the customer profile object's database context and the financial account object's database context must previously have been set by calling each object's <code>createMembershipContext ()</code> method.

ejbStoreFinancialAccount()

Updates a financial account record associated with this customer.

Syntax	<code>void ejbStoreFinancialAccount (IBXFinancialAccount Returns_account) throws Exception;</code>
Parameters	<i>Returns_account</i> — The IBXFinancialAccount object to update in the database
Discussion	Both the customer profile object's database context and the financial account object's database context must previously have been set by calling each object's <code>createMembershipContext ()</code> method.

Important	You must retrieve the record before you update, otherwise fields that you do not explicitly set will be written as null.

getAccount()

Determines the customer's account number.

Syntax	<code>String getAccount ();</code>
Parameters	None.

Discussion	Returns a string that contains the account number.

getAccountType()

Determines the customer's account types.

Syntax	<code>String getAccountType();</code>
Parameters	None.
Discussion	Returns a vector that contains the account type.

getAddress()

Determines the customer's address.

Syntax	<code>IBXAddress getAddress();</code>
Parameters	None.
Discussion	Returns a IBXAddress object that contains the address.

getCharSet()

Determines the customer's character set.

Syntax	<code>String getCharSet();</code>
Parameters	None.
Discussion	Returns a string that contains the character set.

getCreateDate()

Determines the customer profile's creation date.

Syntax	<code>Date getCreateDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the creation date.

getCreatedBy()

Determines the customer profile’s creator.

Syntax	<code>String getCreatedBy() ;</code>
Parameters	None.
Discussion	Returns a string that contains the account’s creator.

getCurrency()

Determines the customer’s currency.

Syntax	<code>String getCurrency() ;</code>
Parameters	None.
Discussion	Returns a string that contains the currency

getCustomerStatus()

Determines the customer’s status.

Syntax	<code>String getCustomerStatus() ;</code>
Parameters	None.
Discussion	Returns a string that contains the customer’s status.

getDayPhone()

Determines the customer’s daytime phone number.

Syntax	<code>String getDayPhone() ;</code>
Parameters	None.
Discussion	Returns a string that contains the phone number.

getDefaultPmtAcct()

Determines the default payment account.

Syntax	<code>String getDefaultPmtAcct();</code>
Parameters	None.
Discussion	Returns a string that contains the default payment account.

getDescription()

Determines the customer's description.

Syntax	<code>String getDescription();</code>
Parameters	None.
Discussion	Returns a string that contains the description..

getEmail()

Determines the customer's email address.

Syntax	<code>String getEmail();</code>
Parameters	None.
Discussion	Returns a string that contains the email address..

getExtension()

Determines the customer's extension.

Syntax	<code>String getExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the extension.

getFaxNumber()

Determines the customer’s fax number.

Syntax	<code>String getFaxNumber();</code>
Parameters	None.
Discussion	Returns a string that contains the number.

getFirstName()

Determines the customer’s given name.

Syntax	<code>String getFirstName();</code>
Parameters	None.
Discussion	Returns a string that contains the name.

getGroup()

Determines the customer’s groups.

Syntax	<code>public Vector getGroup();</code>
Parameters	None.
Discussion	Returns a string that contains the customer’s group(s)..

getLanguage()

Determines the customer’s language.

Syntax	<code>String getLanguage();</code>
Parameters	None.
Discussion	Returns a string that contains the language.

getLastName()

Determines the customer's surname.

Syntax	<code>String getLastName();</code>
Parameters	None.
Discussion	Returns a string that contains the customer's last name.

getLoginName()

Determines the customer's login name.

Syntax	<code>String getLoginName();</code>
Parameters	None.
Discussion	Returns a string that contains the login name.

getMiddleName()

Determines the customer's middle name.

Syntax	<code>String getMiddleName();</code>
Parameters	None.
Discussion	Returns a string that contains the name.

getModifyDate()

Determines the modification date.

Syntax	<code>String getModifyDate();</code>
Parameters	None.
Discussion	Returns a string that contains the modification date.

getNickName()

Determines the customer’s nick name.

Syntax	<code>String getNickName() ;</code>
Parameters	None.
Discussion	Returns a string that contains the name.

getNightPhone()

Determines the customer’s evening phone number.

Syntax	<code>String getNightPhone() ;</code>
Parameters	None.
Discussion	Returns a string that contains the number.

getOrgName()

Determines the customer’s organization name.

Syntax	<code>String getOrgName() ;</code>
Parameters	None.
Discussion	Returns a string that contains the name.

getPassword()

Specifies the customer’s password.

Syntax	<code>String getPassword() ;</code>
Parameters	None.
Discussion	Returns a string that contains the password.

getPasswordHint()

Determines the customer's password hint.

Syntax	<code>String getPasswordHint () ;</code>
Parameters	None.
Discussion	Returns a string that contains the word.

getPasswordLastChanged()

Determines the date the customer's password was last changed.

Syntax	<code>Date getPasswordLastChanged () ;</code>
Parameters	None.
Discussion	Returns a Date object that contains the date.

getPrefix()

Determines the customer's name prefix.

Syntax	<code>String getPrefix () ;</code>
Parameter	None.
Discussion	Returns a string that contains the prefix.

getPrimaryContact()

Determines the customer's primary contact.

Syntax	<code>String getPrimaryContact () ;</code>
Parameters	None.
Discussion	Returns a string that contains the contact.

getProfileExtension()

Determines the customer’s profile extension.

Syntax	<code>String getProfileExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the profile extension.

getSecondaryContact()

Determines the customer’s secondary contacts.

Syntax	<code>Vector getSecondaryContact();</code>
Parameters	None.
Discussion	Returns a vector that contains the contact(s).

getSecretWord()

Determines the customer’s secret word.

Syntax	<code>String getSecretWord();</code>
Parameters	None.
Discussion	Returns a string that contains the word.

getServiceExtension()

Determines the customer’s service extension.

Syntax	<code>String getServiceExtension();</code>
Parameters	None.
Discussion	Returns a string that contains the number.

getSuffix()

Determines the customer's name suffix.

Syntax	<code>String getSuffix();</code>
Parameters	None.
Discussion	Returns a string that contains the suffix.

getTimeZone()

Determines the customer's time zone.

Syntax	<code>String getTimeZone();</code>
Parameters	None.
Discussion	Returns a string that contains the time zone.

setAccount()

Specifies the customer's account number.

Syntax	<code>void setAccount (String value);</code>
Parameters	<i>value</i> — A string that specifies the account
Discussion	This method only changes the value of the account number in the object; it does not modify the value in the database.

setAccountType()

Specifies the customer's account types.

Syntax	<code>void setAccountType (Vector value);</code>
Parameters	<i>value</i> — A vector that specifies the account type
Discussion	This method only changes the value of the account type in the object; it does not modify the value in the database.

setIBXAddress()

Specifies the customer’s address.

Syntax	<code>void setIBXAddress (IBXAddress value) ;</code>
Parameters	<i>value</i> — An IBXAddress object that specifies the address.
Discussion	This method only changes the value of the address field in the object; it does not modify the value in the database.

setAttribute()

Specifies a generic attribute.

Syntax	<code>void setAttribute (String name, String value) ;</code>
Parameters	<i>name</i> — A string that specifies the attribute name. <i>value</i> — A string that specifies the address.
Discussion	This method only changes the value of the attribute field in the object; it does not modify the value in the database.

setCharSet()

Specifies the customer’s character set.

Syntax	<code>void setCharSet (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the character set.
Discussion	This method only changes the value of the character set field in the object; it does not modify the value in the database.

setCreatedBy()

Determines the customer profile’s creator.

Syntax	<code>void setCreatedBy (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the creator.
Discussion	This method only changes the value of the in the creator field in the object; it does not modify the value in the database.

setCurrency()

Specifies the customer's currency.

Syntax	<code>void setCurrency (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the currency
Discussion	This method only changes the value of the currency in the object; it does not modify the value in the database.

setCustomerStatus()

Specifies the customer's status.

Syntax	<code>void setCustomerStatus (String <i>value</i>);</code>
Parameters	<i>value</i> — A string that specifies the status
Discussion	This method only changes the value of the status in the object; it does not modify the value in the database.

setDayPhone()

Specifies the customer's daytime phone number.

Syntax	<code>void setDayPhone (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the daytime phone number
Discussion	This method only changes the value of the phone number in the object; it does not modify the value in the database.

setDefaultPmtAcct()

Specifies the default payment account.

Syntax	<code>void setDefaultPmtAcct (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the default payment account
Discussion	This method only changes the value of the description field in the object; it does not modify the value in the database.

setDescription()

Specifies the customer’s description.

Syntax	<code>void setDescription (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the description
Discussion	This method only changes the value of the description in the object; it does not modify the value in the database.

setEmail()

Specifies the customer’s email address.

Syntax	<code>void setEmail (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the email address
Discussion	This method only changes the value of the email address in the object; it does not modify the value in the database.

setExtension()

Specifies the customer’s extension.

Syntax	<code>void setExtension (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the extension
Discussion	This method only changes the value of the extension in the object; it does not modify the value in the database.

setFaxNumber()

Specifies the customer’s fax number.

Syntax	<code>void setFaxNumber (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the fax number
Discussion	This method only changes the value of the number in the object; it does not modify the value in the database.

setFileID()

Specifies the file ID.

Syntax	<code>void setFileID (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the file ID
Discussion	This method only changes the value of the file ID in the object; it does not modify the value in the database.

setFirstName()

Specifies the customer's given name.

Syntax	<code>void setFirstName (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the name
Discussion	This method only changes the value of the name in the object; it does not modify the value in the database.

setGroup()

Specifies the customer's groups.

Syntax	<code>void setGroup (Vector <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the group(s)
Discussion	This method only changes the value of the group in the object; it does not modify the value in the database.

setLanguage()

Specifies the customer's language.

Syntax	<code>void setLanguage (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the language
Discussion	This method only changes the value of the language in the object; it does not modify the value in the database.

setLastName()

Specifies the customer’s surname.

Syntax	<code>void setLastName (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the name
Discussion	This method only changes the value of the name field the object; it does not modify the value in the database.

setLoginName()

Specifies the customer’s login name.

Syntax	<code>void setLoginName (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the login name
Discussion	This method only changes the value of the login name field in the object; it does not modify the value in the database.

setMiddleName()

Specifies the customer’s middle name.

Syntax	<code>void setMiddleName (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the name
Discussion	This method only changes the value of the middle name field in the object; it does not modify the value in the database.

setNickName()

Specifies the customer’s nick name.

Syntax	<code>void setNickName (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the name
Discussion	This method only changes the value of the nickname field in the object; it does not modify the value in the database.

setNightPhone()

Specifies the customer's evening phone number.

Syntax	<code>void setNightPhone (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the number
Discussion	This method only changes the value of the number field in the object; it does not modify the value in the database.

setOrgName()

Specifies the customer's organization name.

Syntax	<code>void setOrgName (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the name
Discussion	This method only changes the value of the name field in the object; it does not modify the value in the database.

setPassword()

Specifies the customer's password.

Syntax	<code>public void setPassword (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the password
Discussion	This method only changes the value of the password field in the object; it does not modify the value in the database.

setPasswordHint()

Specifies the customer's password hint.

Syntax	<code>void setPasswordHint (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the password hint the customer provided.
Discussion	This method only changes the value of the word field in the object; it does not modify the value in the database.

setPasswordLastChanged()

Specifies the date the customer’s password was last changed.

Syntax	<code>public void setPasswordLastChanged (Date <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the date
Discussion	This method only changes the value of the date field in the object; it does not modify the value in the database.

setPrefix()

Specifies the customer’s name prefix.

Syntax	<code>void setPrefix (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the prefix.
Discussion	This method only changes the value of the prefix field in the object; it does not modify the value in the database.

setPrimaryContact()

Specifies the customer’s primary contact.

Syntax	<code>void setPrimaryContact (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the customer’s primary contact.
Discussion	This method only changes the value of the contact field in the object; it does not modify the value in the database.

setProfileExtension()

Specifies the customer’s profile extension.

Syntax	<code>public void setProfileExtension (String <i>value</i>)</code>
Parameters	<i>value</i> — A string that specifies the profile extension
Discussion	This method only changes the value of the profile extension field in the object; it does not modify the value in the database.

setSecondaryContact()

Specifies the customer's secondary contacts.

Syntax	<code>void setSecondaryContact (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the contact
Discussion	This method only changes the value of the contact field in the object; it does not modify the value in the database.

setSecretWord()

Specifies the customer's secret word.

Syntax	<code>void setSecretWord (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the word
Discussion	This method only changes the value of the word field in the object; it does not modify the value in the database.

setServiceExtension()

Specifies the customer's service extension.

Syntax	<code>void setServiceExtension (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the extension
Discussion	This method only changes the value of the extension field in the object; it does not modify the value in the database.

setSuffix()

Specifies the customer's name suffix.

Syntax	<code>void setSuffix (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the suffix
Discussion	This method only changes the value of the suffix field in the object; it does not modify the value in the database.

setTimeZone()

Specifies the customer’s time zone.

Syntax	<code>void setTimeZone (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the time zone
Discussion	This method only changes the value of the zone field in the object; it does not modify the value in the database.

updatePassword()

Changes the customer’s password.

Syntax	<code>void updatePassword (String loginName, String password) throws Exception;</code>
Parameters	<i>loginName</i> — A string that specifies the login name. <i>value</i> — A string that specifies the word.
Discussion	The <code>updatePassword()</code> method changes the password associated with the specified login name. The organization field must be set in the object and the login name for that organization must exist on the server. This method sets the values of the password, password last changed, and modification date fields in the object and updates the values of these fields in the database.

IBXFinancialAccount Class Reference

The BXFinancialAccount class is an implementation of the IBXFinancialAccount interface, which defines customer profile objects that represent financial account records.

Whenever possible, use the interface class IBXFinancialAccount (see “IBXFinancialAccount Class Reference”) instead of BXFinancialAccount.

IBXFinancialAccount Class Reference

The IBXFinancialAccount interface defines customer profile objects that represent financial account records. You call the `BXMembershipMgr.createFinancialAccount()` method to create a financial Returns account object.

Before you can call the `IBXEntity.ejbCreate()` method to create a new record Returns associated with the object, you must call the `setPayment()` and `setPanHandle()` methods to set the payment handle field.

Interface Definition

Table 3-8 describes the `IBXFinancialAccount` interface definition.

Table 3-8 `IBXFinancialAccount` Interface Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.membership;</code>
Syntax	<code>public class IBXFinancialAccount extends IBXMembershipEntity {...}</code>

Methods

This section describes the `IBXFinancialAccount` interface methods.

Table 3-9 `IBXFinancialAccount` Interface Methods

Method	Description
Getting and setting financial Returns account information	
<code>getPaymentHandle()</code>	Determines the financial account's payment type.
<code>setPaymentHandle()</code>	Specifies the financial account's payment type.
<code>getDescription()</code>	Determines the financial account's description.
<code>setDescription()</code>	Specifies the financial account's description.
<code>getType()</code>	Determines the financial account's payment type.
<code>setType()</code>	Specifies the financial account's payment type.
<code>getAlias()</code>	Determines the user-defined name associated with this financial account.
<code>setAlias()</code>	Specifies the user-defined name associated with this financial account.
<code>getPanHandle()</code>	Determines the PSAL handle associated with this financial account.
<code>setPanHandle()</code>	Specifies the PSAL handle associated with this financial account.
<code>getPaymentHandle()</code>	Determines the financial account's payment type.
<code>getExtension()</code>	Determines the extension associated with this financial account
<code>setExtension()</code>	Specifies the extension associated with this financial account
<code>getRoutingNumber()</code>	Determines the routing number associated with this financial institution.
<code>setRoutingNumber()</code>	Specifies the routing number associated with this financial institution.

Table 3-9 IBXFinancialAccount Interface Methods

Method	Description
getAccountNumber()	Determines the account number associated with this financial institution
setAccountNumber()	Specifies the number associated with this financial institution

getAccountNumber()

Determines the account number.

Syntax	String getAccountNumber();
Parameters	None.
Discussion	Returns a string that contains the account number.

getAlias()

Determines the user-defined name associated with this financial account.

Syntax	String getAlias();
Parameters	None.
Discussion	Returns a string that contains the alias.

getDescription()

Determines the financial account’s description.

Syntax	String getDescription();
Parameters	None.
Discussion	Returns a string that contains the description.

getExtension()

Determines the extension associated with this financial account.

Syntax	String getExtension();
Parameters	None.

Discussion	Returns a string that contains the description.

getPanHandle()

Determines the PSAL handle associated with this financial account.

Syntax	<code>String getPanHandle();</code>

Parameters	None.

Discussion	Returns a string that contains the PSAL handle.

getPaymentHandle()

Determines the financial account's payment type.

Syntax	<code>String getPaymentHandle();</code>

Parameters	None.

Discussion	Returns a string that contains the handle.

getRoutingNumber()

Determines the routing number.

Syntax	<code>String getRoutingNumber();</code>

Parameters	None.

Discussion	Returns a string that contains the routing number.

getType()

Determines the financial account's payment type.

Syntax	<code>String getType();</code>

Parameters	None.

Discussion	Returns a string that contacts the type of payment that will be made.

setAlias()

Specifies the user-defined name associated with this financial account.

Syntax	<code>void setAlias (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the alias
Discussion	This method only changes the value of the alias field in the object; it does not modify the value in the database.

setDescription()

Specifies the financial account’s description.

Syntax	<code>void setDescription (String value) ;</code>
Parameters	<i>value</i> — A string that specifies thedescription
Discussion	This method only changes the value of the description field in the object; it does not modify the value in the database.

setPaymentHandle()

Specifies the financial account’s payment type.

Syntax	<code>void setPaymentHandle (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the handle
Discussion	This method only changes the value of the handle field in the object; it does not modify the value in the database.

setRoutingNumber()

Specifies the financial account’s routing number.

Syntax	<code>void setRoutingNumber (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the routing number
Discussion	This method only changes the value of the routing number field in the object; it does not modify the value in the database.

setType()
 Specifies the financial account’s payment type.

Syntax	<code>void setType (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the type.
Discussion	This method only changes the value of the type field in the object; it does not modify the value in the database.

BXMembershipContext Class Reference

The BXMembershipContext class implements the IBXDBContext interface. Instances of the BXMembershipContext class represent a database context for membership objects.

You typically create a membership object using one of the BXMembershipMgr class methods and then create a context for it by creating a BXMembershipContext object. Because membership objects are not stored in a database per se, the transaction methods are written to throw an exception if called. Because there is no database connection to close, the `release()` method performs no actions.

Class Definition

Table 3-10 describes the BXMembershipContext class definition.

Table 3-10 BXMembershipContext Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.membership;</code>
Syntax	<code>public class BXMembershipContext implements IBXDBContent {...}</code>

Log Message Level Constants

The following constants are used to specify the level associated with log messages for BXMembershipContext class.

ejbAssociateUser()

Returns associates an account with a user.

Syntax	<pre>public static final int ERROR = -1; public static final int WARNING = 0; public static final int INFO = 1;</pre>
ERROR	Error message
WARNING	Warning message
INFO	Informational message

Constructor

This section describes the BXMembershipContext constructor.

Syntax	<pre>BXMembershipContext (IContext context) throws Exception; BXMembershipContext (IContext context, BXStatus status) throws Exception;</pre>
Parameters	<i>context</i> — An IContext object that specifies the context. <i>status</i> — A BXStatus object that holds the current status of the contex
Discussion	Creates a BXMembershipContext object. You typically create a membership context object by specifying the applogic's <i>context</i> variable for the <i>context</i> parameter. The first form of the constructor does not associate a <i>status</i> object with the <i>context</i> . In this case, you must call the <code>setStatus()</code> or <code>getBXStatus()</code> method to specify a status object for the context.

Methods

This section describes the BXMembershipContext methods. Table 3-11 lists these methods.

Table 3-11 BXMembershipContext Methods (1 of 2)

Method	Description
Constructor	
<code>BXMembershipContext()</code>	Creates a BXMembershipContext object.
Getting and setting the status	
<code>getStatus()</code>	Determines the status object currently associated with this entity.

Table 3-11 BXMembershipContext Methods *(2 of 2)*

Method	Description
<code>getBXStatus()</code>	Determines the status object currently associated with this object.
<code>setStatus()</code>	Specifies the status object to associate with this object.
Terminating the context	
<code>release()</code>	Closes the database connection and releases memory for this context object.
<code>finalize()</code>	Cleans up context and releases resources.
Transaction Management	
<code>beginTransaction()</code>	Starts a transaction.
<code>commitTransaction()</code>	Commits a transaction.
<code>rollbackTransaction()</code>	Rolls back a transaction.
Debugging Support	
<code>log()</code>	Writes a message to the log.

beginTransaction()

Starts a transaction.

Syntax	<code>void beginTransaction() throws Exception;</code>
Parameters	None.
Discussion	The <code>beginTransaction()</code> method is not supported for membership objects. This method raises an exception if it is called.

commitTransaction()

Commits a transaction.

Syntax	<code>void commitTransaction() throws Exception;</code>
Parameters	None.
Discussion	Not supported for membership objects. This method raises an exception if it is called.

finalize()

Cleans up context and releases resources.

Syntax	<code>void finalize();</code>
Parameters	None.
Discussion	The <code>finalize()</code> method is called automatically when the object is garbage collected; you do not call this method directly. The <code>finalize()</code> method calls the object's <code>release()</code> method.

getBXStatus()

Determines the status object currently associated with this object.

Syntax	<code>BXStatus getBXStatus();</code>
Parameters	None.
Discussion	If a <code>BXStatus</code> object has not been associated with the context, the <code>getBXStatus()</code> method creates a new <code>BXStatus</code> object and returns a reference to it. The <code>getBXStatus()</code> method is a protected method. Returns a <code>BXStatus</code> object that specifies the current status

getStatus()

Determines the status object currently associated with this entity.

Syntax	<code>BXStatus getStatus();</code>
Parameters	None.
Discussion	Returns a <code>BXStatus</code> object that specifies the current status.

Log()

Writes a message to the log.

Syntax	<code>void log(int severity, String msg);</code>
Parameters	<i>severity</i> — An integer that specifies the message level. <i>msg</i> — A string that contains the message.

Discussion	Use ERROR, WARNING, or INFO for the severity parameter.

release()

Cleans up context and releases resources.

Syntax	<code>void release();</code>
Parameters	None.
Discussion	This method does not perform any action.

rollbackTransaction()

Rolls back a transaction.

Syntax	<code>void rollbackTransaction() throws Exception;</code>
Parameters	None.
Discussion	The <code>rollbackTransaction()</code> method is not supported for membership objects. This method raises an exception if it is called.

setStatus()

Specifies the status object to associate with this object.

Syntax	<code>BXStatus setStatus (BXStatus status);</code>
Parameters	<i>status</i> — A BXStatus object.
Discussion	Returns a reference to the status object.

BXMembershipEntity Class Reference

The `BXMembershipEntity` class is an implementation of the `IBXMembershipEntity` interface, which is the base interface for all membership objects.

Whenever possible, use the interface class `IBXMembershipEntity` (see “`IBXMembershipEntity` Class Reference”) instead of `BXMembershipEntity`.

IBXMembershipEntity Class Reference

The IBXMembershipEntity class is the base interface for all membership objects. You must call the `setMembershipContext()` method of each object that represents Returns a membership record in the database before you can manipulate the record with the object.

Interface Definition

Table 3-12 describes the IBXMembershipEntity interface definition.

Table 3-12 IBXMembershipEntity Interface Definition

Package	com.iplanet.ecomm.billxb2b.ejb.membership;
Syntax	public interface IBXMembershipEntity extend IBXEntity {...}

Methods

This section describes the IBXMembershipEntity methods. Table 3-13 lists these methods.

Table 3-13 IBXMembershipEntity Interface Methods

Methods	Description
getMembershipContext()	Determines the membership context.
setMembershipContext()	Specifies the membership context.

getMembershipContext()

Determines the membership context.

Syntax	BXMembershipContext getMembershipContext();
Parameters	None.
Discussion	The BXMembershipContext object associated with this membership object.

setMembershipContext()

Specifies the membership context.

Syntax	<code>void setMembershipContext (BXMembershipContext context) ;</code>
Parameters	<i>context</i> — The context associated with this membership object.
Discussion	

BXMembershipMgr Class Reference

The BXMembershipMgr class creates membership-related business objects. You typically create a membership object and use it to manipulate a record in a Directory Server database.

Class Definition

Table 3-14 describes the BXMembershipMgr class definition.

Table 3-14 BXMembershipMgr Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.membership;</code>
Syntax	<code>public class BXMembershipMgr { ... }</code>

Methods

This section describes the BXMembershipMgr class methods. Table 3-15 lists these methods.

Table 3-15 BXMembershipMgr Class Methods

Methods	Description
Create membership object	
<code>createAccountType()</code>	Creates an account type object.
<code>createAddress()</code>	Creates an address object.
<code>createBillerProfile()</code>	Creates a billing profile object.
<code>createBillerGroup()</code>	Creates a biller group object
<code>createCustomerProfile()</code>	Creates a customer profile object.

Table 3-15 BXMembershipMgr Class Methods

Methods	Description
<code>createCompanyProfile()</code>	Creates a company profile object.
<code>createDepartment()</code>	Creates a department object.
<code>createFinancialAccount()</code>	Creates a financial account object.
<code>createGroup()</code>	Creates a group object.
<code>createPaymentType()</code>	Creates a payment type object.
Create Entity Object	
<code>createObject()</code>	Creates an object.

createAccountType()

Creates an account type object.

Syntax	<code>static IBXAccountType createAccountType (String <i>billName</i>) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of <code>billerName:acctType</code> .
Discussion	The <code>createAccountType()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBXAccountType object.

createAddress()

Creates an address object.

Syntax	<code>static IBXAccountType createAccountType (String <i>billName</i>) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of <code>billerName:acctType</code> .
Discussion	The <code>createAccountType()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBXAccountType object.

createBillerProfile()

Creates a billing profile object.

Syntax	<code>static IBXBillerProfile createAccountType (String <i>billName</i>) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of billerName:acctType.
Discussion	The <code>createBillerProfile()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBXAccountType object.

createBillerGroup()

Creates a Biller group object.

Syntax	<code>static IBXBillerGroup createBillerGroup (String <i>billName</i>) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of billerName:acctType.
Discussion	Returns an IBXBillerGroup object.

createCompanyProfile()

Creates a company profile object.

Syntax	<code>static IBXCompanyProfile createCompanyProfile (String <i>companyName</i>) throws Exception;</code>
Parameters	<i>companyName</i> — A string that contains the company name
Discussion	Returns an IBXCompanyProfile object.

createCustomerProfile()

Creates a customer profile object.

Syntax	<code>static IBXCustomerProfile createCustomerProfile (String <i>billName</i>) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of billerName:acctType.

Discussion	The <code>createAccountType()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an <code>IBXCustomerProfile</code> object.

createDepartment()

Creates a department object.

Syntax	<code>static IBXDepartment createDepartment(String <i>departmentName</i>)</code> <code>throws Exception;</code>
Parameters	<i>departmentName</i> — A string that contains the department name
Discussion	Returns an <code>IBXDepartment</code> object.

createFinancialAccount()

Creates a financial account object.

Syntax	<code>static IBXFinancialAccount createFinancialAccount(String <i>billName</i>)</code> <code>throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of <code>billerName:acctType</code> .
Discussion	The <code>createFinancialAccount()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an <code>IBXFinancialAccount</code> object.

createGroup()

Creates a group object.

Syntax	<code>static IBXGroup createGroup(String <i>billName</i>)</code> <code>throws</code> <code>Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of <code>billerName:acctType</code> .
Discussion	Returns an <code>IBXGroup</code> object.

createObject()

Creates an object.

Syntax	<code>static IBXEntity createObject(String <i>billName</i>) throws Exception;</code>
Parameters	<i>className</i> — A string that specifies the class to which the object belongs <i>billName</i> — A string that contains the object mapping key in the format of <i>billName:acctType</i> .
Discussion	The <code>createObject()</code> method is a protected class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBXEntity object.

createPaymentType()

Creates a payment type object.

Syntax	<code>static IBXPaymentType createPaymentType(String <i>billName</i>) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of <i>billName:acctType</i>
Discussion	The <code>createPaymentType()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBXPaymentType object.

BXPaymentType Class Reference

The BXPaymentType class is an implementation of the IBXPaymentType interface, which defines customer profile objects that represent payment type records.

Whenever possible, use the interface class IBXPaymentType (see “IBXPaymentType Class Reference”) instead of BXPaymentType.

IBXPaymentType Class Reference

The IBXPaymentType class interface defines customer profile objects that represent payment type records. You call the `BXMembershipMgr.createPaymentType()` method to create a payment type object.

Before you can call the `IBXEntity.ejbCreate()` method to create a new record Returns associated with the object, you must call the `setType()` method to set the type field.

Interface Definition

Table 3-14 describes the `IBXPaymentType` class definition.

Table 3-16 BXMembershipMgr Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.membership;</code>
Syntax	<code>public class BXMembershipMgr {...}</code>

Methods

This section describes the `IBXPaymentType` interface methods.

Table 3-17 IBXPaymentType Interface Methods

Method	Description
Getting and setting payment type information	
<code>getType()</code>	Determines the payment type.
<code>setType()</code>	Specifies the payment type.
<code>getBrand()</code>	Determines the card's brand.
<code>setBrand()</code>	Specifies the card's brand.
<code>getAlias()</code>	Determines the user-defined name associated with this payment type.
<code>setAlias()</code>	Specifies the user-defined name associated with this payment type.
<code>getDescription()</code>	Determines the payment type's description.
<code>setDescription()</code>	Specifies the payment type's description.
<code>getMinAmount()</code>	Determines the minimum payment.
<code>setMinAmount()</code>	Specifies the minimum payment.
<code>getMaxAmount()</code>	Determines the maximum payment.
<code>setMaxAmount()</code>	Specifies the maximum payment.
<code>getProcessor()</code>	Determines the credit card processor.
<code>setProcessor()</code>	Specifies the credit card processor.
<code>getPaymentStatus()</code>	Determines the payment status

Table 3-17 IBXPaymentType Interface Methods

Method	Description
<code>setPaymentStatus()</code>	Specifies the payment status

`getAlias()`

Determines the user-defined name associated with this payment type.

Syntax	<code>String getAlias();</code>
Parameters	None.
Discussion	Returns a string that specifies the alias.

`getBrand()`

Determines the card's brand.

Syntax	<code>String getBrand();</code>
Parameters	None.
Discussion	Brands include: VISA, MasterCard, American Express, Discover, and so forth. Returns a string that specifies the brand.

`getDescription()`

Determines the payment type's description.

Syntax	<code>String getDescription();</code>
Parameters	None.
Discussion	Returns a string that specifies the description.

`getMaxAmount()`

Determines the maximum payment.

Syntax	<code>double getMaxAmount();</code>
Parameters	None.

Discussion	Returns a double precision floating point number that specifies the amount.

getMinAmount()

Determines the minimum payment.

Syntax	<code>double getMinAmount();</code>
Parameters	None.
Discussion	Returns a double precision floating point number that specifies the amount.

getPaymentStatus()

Determines the payment status.

Syntax	<code>String getPaymentStatus();</code>
Parameters	None.
Discussion	Returns a string that contains the payment status

getProcessor()

Determines the credit card processor.

Syntax	<code>String getProcessor();</code>
Parameters	None.
Discussion	Returns a string that specifies the name of the processor.

getType()

Determines the payment type.

Syntax	<code>String getType();</code>
Parameters	None.
Discussion	Returns a string that contains the payment type.

setAlias()

Determines the user-defined name associated with this payment type.

Syntax	<code>void setAlias (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the alias.
Discussion	The method only changes the value of the alias field in the object, it does not modify the value in the database.

setBrand()

Specifies the card's brand.

Syntax	<code>void setBrand (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the brand
Discussion	The method only changes the value of the brand field in the object, it does not modify the value in the database.

setDescription()

Specifies the payment type's description.

Syntax	<code>void setDescription (String value) ;</code>
Parameters	<i>value</i> — A string that specifies the description
Discussion	The method only changes the value of the description field in the object, it does not modify the value in the database.

setMaxAmount()

Specifies the maximum payment.

Syntax	<code>void setMaxAmount (double value) ;</code>
Parameters	<i>value</i> — A double precision floating point number that specifies the amount
Discussion	The method only changes the value of the maximum amount field in the object, it does not modify the value in the database.

setMinAmount()

Specifies the minimum payment.

Syntax	<code>void setMinAmount (double <i>value</i>) ;</code>
Parameters	<i>value</i> — A double precision floating point number that specifies the amount.
Discussion	The method only changes the value of the value field in the object, it does not modify the value in the database.

setPaymentStatus()

Specifies the payment status.

Syntax	<code>void setPaymentStatus (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the payment status
Discussion	The method only changes the value of the payment status field in the object, it does not modify the value in the database.

setProcessor()

Specifies the credit card processor.

Syntax	<code>void setProcessor (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the value
Discussion	The method only changes the value of the field in the object, it does not modify the value in the database.

setType()

Specifies the payment type.

Syntax	<code>void setType (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string that specifies the payment type
Discussion	The method only changes the value of the type field in the object, it does not modify the value in the database.

Event Component Business Object Interfaces

This chapter discusses component business object interface classes for events. These classes provide events and event handlers for handling asynchronous events within BillerXpert.

This chapter includes the following sections:

- Overview
- BXEvent Class Reference Descriptions

Overview

Event classes represent event records in the database. The event server uses these classes to manipulate event-related objects. You can use these classes to get and set fields in the objects. Table 4-1 lists these classes.

Table 4-1 Event Classes (1 of 2)

Class Name	Description	Location
1. BXEvent	Base class that manipulates event objects.	page 386
2. IBXEvent	Interface class that manipulates event objects.	page 386
3. BXEventContext	Establishes a context between an AppLogic and a billing component business object.	page 391
4. BXEventEntity	Base class that manipulates event-related objects.	page 392
5. IBXEventEntity	Interface class that manipulates event-related objects.	page 393
6. BXEventHandler	Base class that manipulates event handler mapping objects.	page 394

Table 4-1 Event Classes (2 of 2)

Class Name	Description	Location
7. IBXEventHandler	Interface class that manipulates event handler mapping objects.	page 394
8. BXEventMgr	Creates event-related component business objects.	page 401
9. BXEventQueue	Base class that manipulates event queue objects.	page 403
10. IBXEventQueue	Interface class that manipulates event queue objects.	page 403
11. BXHandler	Base class that manipulates handler objects.	page 412
12. IBXHandler	Interface class that manipulates handler objects.	page 412

BXEvent Class Reference

The BXEvent base class implements the IBXEvent interface, which defines objects that represent event records.

Whenever possible, use the interface class IBXEvent (see “IBXEvent Class Reference) instead of BXEvent.

IBXEvent Class Reference

The IBXEvent interface defines methods that manipulate objects representing event records. To create an event object, call the BXEventMgr.createEvent() method (see page 401) to create an event object.

Before you can call the IBXEntity.IBXEntity.ejbCreate() method (see page 129) to create a new record associated with the object, you must call the setEventName() method (see page 390) to set the event name.

Interface Definition

Table 4-2 describes the IBXEvent interface definition.

Table 4-2 IBXEvent Class Definition

Package	com.iplanet.ecomm.billxb2b.ejb.event;
Syntax	public interface IBXEvent extends IBXEventEntity {...}

Methods

This section describes the IBXEvent class methods. Table 4-3 lists these methods.

Table 4-3 IBXEvent Class Methods

Methods	Description
Manipulating customer account and related records	
<code>ejbFindAll()</code>	
<code>ejbFindByEventName()</code>	Retrieves an event record by event name.
<code>ejbRemoveAll()</code>	Removes all specified events. Getting and setting event information
<code>getEventDesc()</code>	Determines the event description.
<code>setEventDesc()</code>	Specifies the event description.
<code>getEventID()</code>	Determines the event ID.
<code>setEventID()</code>	Specifies the event ID.
<code>getEventName()</code>	Determines the event name.
<code>setEventName()</code>	Specifies the event name
<code>getBillerID()</code>	Determines the Biller ID.
<code>setBillerID()</code>	Specifies the Biller ID
<code>getEventStatus()</code>	Determines the event status.
<code>setEventStatus()</code>	Specifies the event status

`ejbFindAll()`

Retrieves the biller name.

Syntax	<code>public SearchResultSet ejbFindAll (BXFilter <i>filter</i>, BXOrder <i>order</i>, int <i>startIndex</i>, int <i>length</i>, int <i>maxRows</i>) throws Exception;</code>
Parameters	<i>filter</i> — A BXFilter object that specifies the filter. <i>order</i> — A BXOrder object that specifies the search order. <i>startindex</i> — An integer that indicates where the search starts. <i>length</i> — An integer that indicates the length. <i>maxRows</i> — An integer that indicates the maximum number of rows.
Discussion	

ejbFindByEventName()

Retrieves an event record by event name.

Syntax	<code>public void ejbFindByEventName (String <i>EventName</i>) throws Exception;</code>
Parameters	<i>EventName</i> — A string that specifies the event name.
Discussion	Retrieves the event record specified by the event name. The event’s database context must previously have been set by calling this object’s <code>setEventContext ()</code> method. The values of fields already set in the object are overwritten by this method; values in the database are not affected.

ejbRemoveAll()

Removes all specified events.

Syntax	<code>public void ejbRemoveAll (BXFilter <i>filter</i>) throws Exception;</code>
Parameters	<i>filter</i> — A BXFilter object that specifies the filter.
Discussion	Removes the event records specified by the filter parameter; if null is specified, all events are deleted. The event’s database context must previously have been set by calling this object’s <code>setEventContext ()</code> method. The <code>ejbRemoveAll ()</code> method can be committed or rolled back as part of a transaction.

getEventDesc()

Determines the event description.

Syntax	<code>public String getEventDesc ();</code>
Parameters	None.
Discussion	Returns a string that contains the description.

getEventID()

Determines the event ID.

Syntax	<code>public String getEventID ();</code>

Parameters	None.
Discussion	Returns a string that contains the ID.

getEventID()

Determines the event name.

Syntax	<code>public String getEventID();</code>
Parameters	None.
Discussion	Returns a string that contains the event ID.

getEventName()

Determines the event name.

Syntax	<code>public String getEventName ();</code>
Parameters	None.
Discussion	Returns a string that contains the name.

getBillerID()

Determines the event name.

Syntax	<code>public String getBillerID ();</code>
Parameters	None.
Discussion	Returns a string that contains the Biller ID.

getEventStatus()

Determines the event name.

Syntax	<code>public String getEventStatus ();</code>
Parameters	
Discussion	Returns a string that contains the event status.

setEventDesc()

Specifies the event description.

Syntax	<code>public void setEventDesc (String <i>eventDesc</i>)</code>
Parameters	<i>eventDesc</i> — A string that specifies the description.
Discussion	This method only changes the value of the event description in the object; it does not modify the value in the database.

setEventID()

Specifies the event ID.

Syntax	<code>public void setEventID (String <i>eventID</i>)</code>
Parameters	<i>eventID</i> — A string that specifies the ID.
Discussion	This method only changes the value of the event ID in the object; it does not modify the value in the database.

setEventName()

Determines the event name.

Syntax	<code>public void setEventName (String <i>eventName</i>)</code>
Parameters	<i>eventName</i> — A string that specifies the name
Discussion	This method only changes the value of the event name in the object; it does not modify the value in the database.

setBillerID()

Determines the event name.

Syntax	<code>public void setBiller ID (String <i>billerID</i>)</code>
Parameters	<i>billerID</i> — A string that specifies the biller ID
Discussion	This method only changes the value of the biller ID in the object; it does not modify the value in the database.

setStatus()

Determines the event name.

Syntax	<code>public void setStatus (String status)</code>
Parameters	<i>status</i> — A string that specifies the status
Discussion	This method only changes the value of the status in the object; it does not modify the value in the database.

BXEventContext Class Reference

The BXEventContext class implements the IBXDBContext interface. Instances of the BXEventContext class represent a database context for event objects. You typically create an object using one of the BXEventMgr class methods and then create a context for it by creating a BXEventContext object.

Class Definition

Table 4-4 describes the BXEventContext class definition.

Table 4-4 BXEventContext Class Description

Package	<code>com.iplanet.ecomm.billxb2b.ejb.event</code>
Syntax	<code>public class BXEventContext extends BXDBContext {...}</code>

Constructors

The BXEvent Class contains the following constructor:

Syntax	<pre>public BXEventContext {IContext context} throws Exception; public BXEventContext {IContext context, BXStatus status} throws Exception; public BXEventContext {IContext context, String datasource, String database, String username, String passwd, int trace};</pre>
Parameters	<i>context</i> — An IContext object that specifies the context. <i>status</i> — A BXStatus object that holds the current status of the context <i>datasource</i> — A string that specifies the data source <i>database</i> — A string that specifies the database <i>username</i> — A string that specifies the user name <i>passwd</i> — A string that specifies the password.
Discussion	Creates a BXEventContext object. You typically create an event context object by specifying the applogic's <i>context</i> variable for the <i>context</i> parameter. The first and third forms of the constructor do not associate a status object with the context. In this case, you must call the <code>setStatus()</code> or <code>getBXStatus()</code> method to specify a status object for the context. The first two forms of the constructor take the database login information from a configuration file. If you use the third form of the constructor, specify the same value for the <i>datasource</i> parameter as you do for the <i>database</i> parameter.

BXEventEntity Class Reference

The BXEventEntity class is an implementation of IBXEventEntity interface, which is the base interface for all event objects.

Whenever possible, use the interface class IBXEventEntity (see “BXEventHandler Class Reference”) instead of BXAddress.

IBXEventEntity Class Reference

The IBXEventEntity class is the base interface for all event objects. You must call the `setEventContext()` method of each object that represents a billing record in the database before you can manipulate the record with the object.

Interface Definition

Table 4-5 describes the IBXEventEntity interface description.

Table 4-5 IBXEventEntity Interface Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.event</code>
Syntax	<code>public interface IBXEventEntity extends IBXEntity {...}</code>

Methods

This section describes the IBXEventEntity interface methods. Table 4-6 lists these methods.

Table 4-6 IBXEventEntity Class Methods

Methods	Description
Getting and setting event context information	
<code>getEventContext()</code>	Determines the event context.
<code>setEventContext()</code>	Specifies the event context.

getEventContext()

Determines the event context.

Syntax	<code>public BXEventContext getEventContext ();</code>
Parameters	None.
Discussion	Returns the BXEventContext object associated with this event object.

setBillingContext()

Determines the event context.

Syntax	<code>public void setEventContext (BXEventContext context) ;</code>
Parameters	<i>context</i> — The context associated with this billing object
Discussion	

BXEventHandler Class Reference

The BXEventHandler class is an implementation of IBXEventHandler interface, which defines objects that represent event handler records.

Whenever possible, use the interface class IBXEventHandler (see “IBXEventHandler Class Reference”) instead of BXEventHandler.

IBXEventHandler Class Reference

The IBXEventHandler class defines objects that represent event handler records. You call the `BXEventMgr.createEventHandler()` method to create an event handler object.

Before you can call the `IBXEntity.ejbCreate()` method to create a new record associated with the object, you must call the `setEventName()`, `setEventHandler()`, and `setEventHandlerStatus()` methods to set the event name, event handler name, and event handler status, respectively.

Class Description

Table 4-7 describes the IBXEventHandler class description.

Table 4-7 IBXEventHandler Class Description

Package	<code>com.iplanet.ecomm.billxb2b.ejb.event</code>
Syntax	<code>public interface IBXEvent extends IBXEventEntity {...}</code>

Methods

Table 4-8 describes the IBXEventHandler class methods.

Table 4-8 IBXEventHandler Class Methods

Methods	Description
Manipulating customer account and related records	
<code>ejbFindByEventName()</code>	Retrieves an event handler record by event name.
<code>ejbRemoveAll()</code>	Removes all specified event handlers.
Getting and setting credit card information	
<code>getBillerID()</code>	Determines the biller ID associated with this handler.
<code>setBillerID()</code>	Specifies the biller ID associated with this handler.
<code>getEventHandler()</code>	Determines the event handler's class name.
<code>setEventHandler()</code>	Specifies the event handler's class name.
<code>getEventHandlerID()</code>	Determines the event handler ID.
<code>setEventHandlerID()</code>	Specifies the event handler ID.
<code>getEventName()</code>	Determines the event name.
<code>setEventName()</code>	Specifies the event name.
<code>getExtension1()</code>	Determines extension 1
<code>setExtension1()</code>	Specifies extension 1
<code>getExtension 2()</code>	Determines extension 2
<code>setExtension 2()</code>	Specifies extension 2
<code>getExtension3()</code>	Determines extension 3
<code>setExtension3()</code>	Specifies extension 3
<code>getExtension4()</code>	Determines extension 4
<code>setExtension4()</code>	Specifies extension 4

ejbFindByEventName()

Retrieves an event handler record by event name.

Syntax	<pre>public void ejbFindByEventName (String <i>EventName</i>) throws Exception; public void ejbFindByEventName (String <i>EventName</i>, String <i>BillerID</i>) throws Exception;</pre>
Parameters	<i>EventName</i> — A string that specifies the event name <i>BillerID</i> — A string that specifies the event name
Discussion	The <code>ejbFindByEventName()</code> method retrieves the event handler record specified by the event name. The fields of this object are set from the contents of the record. The event handler's database context must previously have been set by calling this object's <code>setEventContext()</code> method. The values of fields already set in the object are overwritten by this method; values in the database are not affected.

ejbRemoveAll()

Removes all specified event handlers.

Syntax	<pre>public void ejbRemoveAll (BXFilter <i>filter</i>) throws Exception;</pre>
Parameters	<i>filter</i> — A <code>BXFilter</code> object that specifies the filter
Discussion	The <code>ejbRemoveAll()</code> method removes the event handler records specified by the filter parameter; if null is specified, all event handlers are deleted. The event handler's database context must previously have been set by calling this object's <code>setEventContext()</code> method. The <code>ejbRemoveAll()</code> method can be committed or rolled back as part of a transaction.

getBillerID

Determines the biller ID associated with this handler.

Syntax	<pre>public String getBillerID ();</pre>
Parameters	None.
Discussion	Returns a string that contains the ID.

getEventHandler()

Determines the event handler's class name.

Syntax	<code>public String getEventHandler ();</code>
Parameters	None.
Discussion	Returns a string that contains the handler's class name.

getEventHandlerStatus()

Determines the event handler status.

Syntax	<code>public String getEventHandlerStatus ();</code>
Parameters	None.
Discussion	Returns a string that contains the status.

getEventName()

Specifies the event name.

Syntax	<code>public String getEventName ();</code>
Parameters	None.
Discussion	Returns a string that contains the name.

getExtension1()

Specifies extension one.

Syntax	<code>public String getExtension1 ();</code>
Parameters	None.
Discussion	Returns a string that contains extension 1.

getExtension2()

Specifies extension one.

Syntax	<code>public String getExtension2();</code>
Parameters	None.
Discussion	Returns a string that contains extension 2.

getExtension3()

Specifies extension one.

Syntax	<code>public String getExtension3();</code>
Parameters	None.
Discussion	Returns a string that contains extension 3.

getExtension4()

Specifies extension one.

Syntax	<code>public String getExtension4 ();</code>
Parameters	None.
Discussion	Returns a string that contains extension 4.

setBillerID()

Specifies the biller ID associated with this handler.

Syntax	<code>public void setBillerID (String <i>billerID</i>);</code>
Parameters	<i>billerID</i> — A string that specifies the ID
Discussion	This method only changes the value of the event biller ID in the object; it does not modify the value in the database.

setEventHandler()

Specifies the event handler's class name.

Syntax	<code>public void setEventHandler (String <i>eventHandler</i>) ;</code>
Parameters	<i>eventHandler</i> — A string that specifies the event handler's name.
Discussion	This method only changes the value of the event handler in the object; it does not modify the value in the database.

setEventHandlerStatus()

Specifies the event handler status.

Syntax	<code>public void setEventHandlerStatus (String <i>status</i>) ;</code>
Parameters	<i>status</i> — A string that specifies the status
Discussion	This method only changes the value of the status in the object; it does not modify the value in the database.

setBillerName()

Specifies the biller name.

Syntax	<code>public void setBillerName (String <i>value</i>) ;</code>
Parameters	<i>value</i> — A string representing the biller name.
Discussion	

setEventName()

Specifies the event name.

Syntax	<code>public void setEventName (String <i>eventName</i>) ;</code>
Parameters	<i>eventName</i> — A string that specifies the event name .
Discussion	This method only changes the value of the event name in the object; it does not modify the value in the database.

setExtension1()

Specifies extension 1.

Syntax	<code>public void setExtension1 (String <i>extension1</i>) ;</code>
Parameters	<i>extension1</i> — A string that specifies extension 1.
Discussion	This method only changes the value of extension1 in the object; it does not modify the value in the database.

setExtension2()

Specifies extension 2.

Syntax	<code>public void setExtension2 (String <i>extension2</i>) ;</code>
Parameters	<i>extension2</i> — A string that specifies extension 1
Discussion	This method only changes the value of extension2 in the object; it does not modify the value in the database.

setExtension3()

Specifies extension 3.

Syntax	<code>public void setExtension3 (String <i>extension3</i>) ;</code>
Parameters	<i>extension3</i> — A string that specifies extension 1
Discussion	This method only changes the value of extensio3 in the object; it does not modify the value in the database.

setExtension4()

Specifies extension 4.

Syntax	<code>public void setExtension4 (String <i>extension4</i>) ;</code>
Parameters	<i>extension4</i> — A string that specifies extension 1
Discussion	This method only changes the value of extension4 in the object; it does not modify the value in the database.

BXEventManager Class Reference

The `BXBillingMgr` class defines methods that create membership objects. You typically create a membership object and use it to manipulate a record in the database.

Class Definition

Table 4-9 describes the `BXEventManager` class definition.

Table 4-9 BXEventManager Class Definition	
Package	<code>com.iplanet.ecomm.billxb2b.ejb.event</code>
Syntax	<code>public class BXEventManager { ... }</code>

Methods

Table 4-10 describes the `BXEventManager` class methods.

Table 4-10 BXEventManager Class Methods	
Methods	Description
Create Billing objects	
<code>createEvent()</code>	Creates an event object.
<code>createEventHandler()</code>	Creates an event handler object.
<code>createEventQueue()</code>	Creates an event queue object.
<code>createHandler()</code>	Creates a handler object.
Create entity object	
<code>createObject()</code>	Creates an object.

createEvent()

Creates an event object.

Syntax	<code>public static IBXEvent createEvent (String <i>billName</i>) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of <code>billName.acctType</code> .

Discussion	The <code>createEvent()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBXEvent object;

createEventHandler()

Creates an event handler object.

Syntax	<code>public static IBXEventHandler createEventHandler (String billName) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of <code>billName.acctType</code> .
Discussion	The <code>createEventHandler()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBXEventHandler object.

createEventQueue()

Creates an event queue object.

Syntax	<code>public static IBXEventQueue createEventHandler (String billName) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of <code>billName.acctType</code> .
Discussion	The <code>createEventQueue()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBXEventQueue object.

createHandler()

Creates a handler object.

Syntax	<code>public static IBXHandler createHandler (String billName) throws Exception;</code>
Parameters	<i>billName</i> — A string that contains the object mapping key in the format of <code>billName.acctType</code> .
Discussion	The <code>createHandler()</code> method is a class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBXHandler object.

createObject()

Creates an object.

Syntax	<code>static IBXEntity createObject (String className, String billName) throws Exception;</code>
Parameters	<i>className</i> — A string that specifies the class to which the object belongs <i>billName</i> — A string that contains the object mapping key in the format of <i>billerName.acctType</i> .
Discussion	The <code>createObject ()</code> method is a protected class method. If you use only the default account, specify null for the <i>billName</i> parameter. Returns an IBEntity object.

BXEventQueue Class Reference

The BXEventQueue class is an implementation of IBXEventHandler interface, which defines objects that represent event queue records.

Whenever possible, use the interface class IBXEventHandler (see “IBXEventQueue Class Reference”) instead of BXEventQueue.

IBXEventQueue Class Reference

The IBXEventQueue class defines objects that represents event queue records. You call the `BXEventManager.createEventQueue ()` method to create an event queue object.

Before you can call the `IBXEntity.ejbCreate ()` method to create a new record associated with the object, you must call the `setEventName ()` and `setEventData ()` methods to set the event name and event data, respectively.

Interface Definition

Table 4-6 describes the IBXEventQueue interface definition.

Table 4-11 IBXEventQueue Interface Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.event</code>
Syntax	<code>public interface IBXEventQueue extends IBXEventEntity {...}</code>

Methods

This section describes the IBXEventQueue interface methods. Table 4-12 lists these methods.

Table 4-12 IBXEventQueue Methods (1 of 2)

Methods	Description
Manipulating customer account and related records	
<code>ejbFindByEventID()</code>	Retrieves an event queue record by event ID.
<code>ejbXFindAll()</code>	Retrieves event queue records. Getting and setting event information
<code>ejbGetUniqueEventIDFromQueue()</code>	Determines an unique event ID from the event queue
<code>ejbGetEventIDsFromQueue()</code>	Determines event IDs from the event queue
Getting and setting event information	
<code>getBillerID()</code>	Determines the biller ID associated with this event queue entry.
<code>setBillerID()</code>	Specifies the biller ID associated with this handler.
<code>getEventData()</code>	Determines the event queue element's data.
<code>setEventData() S</code>	pecifies the event queue element's data.
<code>getEventName()</code>	Determines the event name.
<code>setEventName()</code>	Specifies the event name.
<code>getEventPopDate()</code>	Determines the date the entry was logically removed from queue.
<code>setEventPopDate()</code>	Specifies the date the entry was logically removed from the queue.
<code>getEventPriority()</code>	Determines the event's priority.
<code>setEventPriority()</code>	Specifies the event's priority.
<code>getEventProcessDate()</code>	Determines the date the entry was processed.
<code>setEventProcessDate()</code>	Specifies the date the entry was processed.
<code>getEventPushDate()</code>	Determines the date the entry was added to the queue.
<code>setEventPushDate()</code>	Specifies the date the entry was added to the queue.
<code>getEventStatus()</code>	Determines the event queue entry's status.
<code>setEventStatus()</code>	Specifies the event queue entry's status.
<code>getEventQueueID()</code>	Determines the event queue ID.

Table 4-12 IBXEventQueue Methods (2 of 2)

Methods	Description
setEventQueueID()	Specifies the event queue ID.
getEventDate()	Determines the event date.
setEventDate()	Specifies the event date.
getEventScheduleID()	Determines the event schedule ID
setEventScheduleID()	Specifies the event schedule ID.

ejbFindByEventID()

Retrieves an event handler record by event name.

Syntax	<code>public void ejbFindByEventName (String <i>EventID</i>) throws Exception;</code>
Parameters	<i>EventID</i> — A string that specifies the event ID.
Discussion	The <code>ejbFindByEventID()</code> method retrieves the event queue record specified by the <i>eventID</i>. The fields of this object are set from the contents of the record. The event queue object’s database context must previously have been set by calling this object’s <code>setEventContext()</code> method. The values of fields already set in the object are overwritten by this method; values in the database are not affected.

ejbXFindAll()

Retrieves event queue records.

Syntax	<code>public Vector ejbXFindAll (BXFilter <i>filter</i>, BXOrder <i>order</i>) throws Exception;</code>
Parameters	<i>filter</i> — A BXFilter that specifies the filter. <i>order</i> — A string that specifies the order of the event queue records.
Discussion	A vector that contains BXEventQueue objects representing the selected records.

Discussion	The <code>ejbXFindAll()</code> method retrieves event queue records that have not been logically removed from the queue. The event queue object's database context must previously have been set by calling each object's <code>createEventContext()</code> method. For each record, the <code>ejbXFindAll()</code> method creates an object, sets the fields in the object from a retrieved record, and adds the object to the return vector.

ejbGetUniqueEventIDsFromQueue()

Retrieves unique event IDs from the event queue.

Syntax	<code>public void ejbGetUniqueEventIDsFromQueue() throws Exception;</code>
Parameters	None.
Discussion	The <code>ejbGetUniqueEventIDsFromQueue()</code> method retrieves the event queue record specified. The fields of this object are set from the contents of the record. The event queue object's database context must previously have been set by calling this object's <code>setEventContext()</code> method. The values of fields already set in the object are overwritten by this method; values in the database are not affected. Returns a vector that contains BXEventQueue objects representing the selected records.

ejbGetEventQueueIDsFromQueue()

Retrieves event queue IDs from the event queue.

Syntax	<code>public void ejbGetEventQueueIDsFromQueue() throws Exception;</code>
Parameters	None.
Discussion	The <code>ejbGetEventQueueIDsFromQueue()</code> method retrieves event IDs from the event queue. The fields of this object are set from the contents of the record. The event queue object's database context must previously have been set by calling this object's <code>setEventContext()</code> method. The values of fields already set in the object are overwritten by this method; values in the database are not affected. Returns a vector that contains BXEventQueue objects representing the selected records.

getBillerID()

Determines the biller ID associated with this event queue entry.

Syntax	<code>public String getBillerID();</code>
Parameters	None.
Discussion	Returns a string that contains the ID.

getEventData()

Determines the event queue element's data.

Syntax	<code>public String getEventData();</code>
Parameters	None.
Discussion	Returns a string that contains the name.

getEventPopDate()

Determines the date the entry was logically removed from the queue.

Syntax	<code>public Date getEventPopDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the date.

getEventPriority()

Determines the event's priority.

Syntax	<code>public String getEventName();</code>
Parameters	None.
Discussion	Returns a string that contains the priority.

getEventProcessDate()

Determines the date the entry was processed.

Syntax	<code>public Date getEventPopDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the date.

getEventPushDate()

Determines the date the entry was added to the queue.

Syntax	<code>public Date getEventPopDate();</code>
Parameters	None.
Discussion	Returns a Date object that contains the date.

getEventStatus()

Determines the event queue entry’s status.

Syntax	<code>public String getEventStatus();</code>
Parameters	None.
Discussion	Returns a string that contains the status.

getEventQueueID()

Determines the event queue id.

Syntax	<code>public String getEventQueueID();</code>
Parameters	None.
Discussion	Returns a string that contains the event queue ID.

getEventDate()

Determines the event queue entry's status.

Syntax	<code>public Date getEventDate();</code>
Parameters	None.
Discussion	Returns a Date that contains the event's date.

getEventScheduleID()

Determines the event schedule ID.

Syntax	<code>public String getEventScheduleID();</code>
Parameters	None.
Discussion	Returns a string that contains the event schedule ID.

setBillerID()

Specifies the biller ID associated with this handler.

Syntax	<code>public void setBillerID (String <i>billerID</i>);</code>
Parameters	<i>billerID</i> — A string that specifies the ID.
Discussion	This method only changes the value of the event biller ID in the object; it does not modify the value in the database

setEventData()

Specifies the event queue element's data.

Syntax	<code>public void setEventData (String <i>eventData</i>);</code>
Parameters	<i>eventData</i> — A string that specifies the data
Discussion	This method only changes the value of the event data in the object; it does not modify the value in the database.

setEventID()

Specifies the event name.

Syntax	<code>public void setEventID (String <i>eventID</i>) ;</code>
Parameters	<i>eventID</i> — A string that specifies the event ID
Discussion	This method only changes the value of the event ID in the object; it does not modify the value in the database.

setEventPopDate()

Specifies the date the entry was logically removed from the queue.

Syntax	<code>public void setEventPopDate (Date <i>popDate</i>) ;</code>
Parameters	<i>popDate</i> — A Date object that specifies the date.
Discussion	This method only changes the value of the pop date in the object; it does not modify the value in the database.

setEventPriority()

Specifies the event's priority.

Syntax	<code>public void setEventPriority (String <i>eventPriority</i>) ;</code>
Parameters	<i>eventPriority</i> — A string that specifies the priority
Discussion	

setEventProcessDate()

Specifies the date the entry was processed.

Syntax	<code>public void setEventProcessDate (Date <i>processDate</i>) ;</code>
Parameters	<i>processDate</i> — A Date object that specifies the date
Discussion	This method only changes the value of the process date in the object; it does not modify the value in the database.

setEventPushDate()

Specifies the date the entry was added to the queue.

Syntax	<code>public void setEventPushDate (Date <i>pushDate</i>) ;</code>
Parameters	<i>pushDate</i> — A Date object that specifies the date.
Discussion	This method only changes the value of the push date in the object; it does not modify the value in the database.

setEventStatus()

Specifies the event queue entry's status.

Syntax	<code>public void setEventStatus (String <i>status</i>) ;</code>
Parameters	<i>status</i> — A string that specifies the status
Discussion	This method only changes the value of the status in the object; it does not modify the value in the database.

setEventQueueID()

Specifies the event queue ID.

Syntax	<code>public void setEventQueueID (String <i>eventQueueID</i>) ;</code>
Parameters	<i>eventQueueID</i> — A string that specifies the event queue ID
Discussion	This method only changes the value of the event queue ID in the object; it does not modify the value in the database.

setEventDate()

Specifies the event date.

Syntax	<code>public void setEventDate (Date <i>date</i>) ;</code>
Parameters	<i>date</i> — A Date object that specifies the event date
Discussion	This method only changes the value of the event date in the object; it does not modify the value in the database.

setEventScheduleID()

Specifies the event schedule ID.

Syntax	<code>public void setEventScheduleID (String eventScheduleID) ;</code>
Parameters	<i>eventScheduleID</i> — A string that specifies the event schedule ID
Discussion	This method only changes the value of the event schedule ID in the object; it does not modify the value in the database.

BXHandler Class Reference

The BXHandler class is an implementation of IBXEventHandler interface, which defines objects that represent handler records.

Whenever possible, use the interface class IBXEventHandler (see “IBXHandler Class Reference”) instead of BXHandler.

IBXHandler Class Reference

The IBXHandler class defines objects that represent handler records. A handler record defines the applogic that responds to an event and other information related to the applogic’s environment. This class also defines methods that allow you to locate the handler by name and to remove handlers. You call the `BXEventManager.createHandler()` method to create a handler object.

Before you can call the `IBXEntity.ejbCreate()` method to create a new record associated with the object, you must call the `setHandlerName()` and `setApplicationName()` methods to set the handler name and application name, respectively.

Class Definition

Table 4-13 describes the IBXHandler class definition.

Table 4-13 IBXHandler Class Definition

Package	<code>com.iplanet.ecomm.billxb2b.ejb.event</code>
Syntax	<code>public interface IBXHandler extends IBXEventEntity {...}</code>

Methods

Table 4-14 describes the IBXHandler class methods.

Table 4-14 IBXHandler Class Methods

Methods	Description
Manipulating customer account and related records	
<code>ejbFindByHandlerName()</code>	Retrieves an event handler record by handler name.
<code>ejbRemoveAll()</code>	Removes all specified handlers.
Getting and setting handler information	
<code>getApplicationName()</code>	Determines the applogic ID associated with this handler..
<code>setApplicationName()</code>	Specifies the applogic ID associated with this handler.
<code>getHandlerClassPath()</code>	Determines the handling applogic's class path.
<code>setHandlerClassPath()</code>	Specifies the handling applogic's class path.
<code>getHandlerDesc()</code>	Determines the handler's description.
<code>setHandlerDesc()</code>	Specifies the handler's description.
<code>getHandlerID()</code>	Determines the handler ID.
<code>setHandlerID()</code>	Specifies the handler ID.
<code>getHandlerName()</code>	Determines the handler's name.
<code>setHandlerName()</code>	Specifies the handler's name.
<code>getHandlerPackage()</code>	Determines the handling applogic's package.
<code>setHandlerPackage()</code>	Specifies the handling applogic's package.

`ejbFindByHandlerName()`

Retrieves an event handler record by handler name.

Syntax	<code>public void ejbFindByHandlerName (String <i>HandlerName</i>) throws Exception;</code>
Parameters	<i>HandlerName</i> — A string that specifies the handler name.

Discussion	The <code>ejbFindByHandlerName()</code> method retrieves the handler record specified by the handler name. The fields of this object are set from the contents of the record. The handler's database context must previously have been set by calling this object's <code>setEventContext()</code> method. The values of fields already set in the object are overwritten by this method; values in the database are not affected.

ejbRemoveAll()

Removes all specified handlers.

Syntax	<code>public void ejbRemoveAll (BXFilter <i>filter</i>) throws Exception;</code>
Parameters	<i>filter</i> — A BXFilter object that specifies the filter.
Discussion	The <code>ejbRemoveAll()</code> method removes the handler records specified by the filter parameter; if null is specified, all handlers are deleted. The handler's database context must previously have been set by calling this object's <code>setEventContext()</code> method. The <code>ejbRemoveAll()</code> method can be committed or rolled back as part of a transaction

getApplicationName()

Determines the applogic ID associated with this handler.

Syntax	<code>public String getApplicationName();</code>
Parameters	None.
Discussion	Returns a string that contains the ID

getHandlerClassPath()

Determines the handling AppLogic's class path.

Syntax	<code>public String getHandlerClassPath();</code>
Parameters	None.
Discussion	Returns a string that contains the class path.

getHandlerDesc()

Determines the handler's description.

Syntax	<code>public String getHandlerDesc();</code>
Parameters	None.
Discussion	Returns a string that contains the handler's description.

getHandlerID()

Determines the handler ID.

Syntax	<code>public String getHandlerID();</code>
Parameters	None.
Discussion	Returns a string that contains the ID

getHandlerName()

Determines the handler's name.

Syntax	<code>public String getHandlerName();</code>
Parameters	None.
Discussion	Returns a string that contains the name.

getHandlerPackage()

Determines the handling applogic's package.

Syntax	<code>public String getHandlerPackage();</code>
Parameters	None.
Discussion	Returns a string that contains the package name.

setApplicationName()

Specifies the AppLogic ID associated with this handler.

Syntax	<code>public void setApplicationName (String <i>applicationName</i>) ;</code>
Parameters	<i>applicationName</i> — A string that specifies the applogic ID.
Discussion	This method only changes the value of the applogic ID in the object; it does not modify the value in the database.

BillerXpert Database Schemas

Chapter 5, “Billing Schema”

Chapter 6, “Membership Schema”

Chapter 7, “Payment Tables”

Chapter 8, “Currency Conversion Tables”

Billing Schema

This chapter describes the structure of the Oracle tables BillerXpert uses to handle billing.

This chapter includes the following sections:

- Billing Schema
- Relationship Between Tables
- Listing of Tables
- Event Tables
- Activity Tracking Tables
- Document Tables
- Payment Related Tables
- Assignment Tables
- Other Tables

CAUTION BillerXpert provides an API that you should use to access the database instead of the schema. The database schema for BillerXpert is subject to change in future versions of BillerXpert. If you rely on the schema, you should consider the potential reimplementing effort that you could incur as the result of an upgrade to the database.

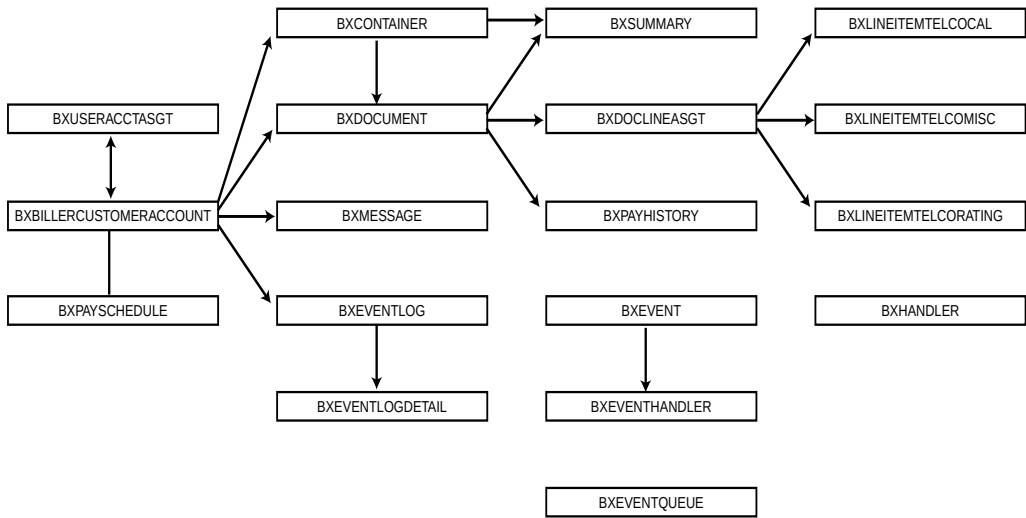
Billing Schema

The billing database schema consists of tables that represent customer account and statement information. It also includes tables on event and event handling information and relationship information between customers and accounts. The billing table listing provides a list of all the billing database schema tables and a short description of each table. Refer to a table’s referenced page number for more information about that table.

Relationship Between Tables

Figure 5-1 shows the relationship between the Billing tables.

Figure 5-1 Billing table relationships



Listing of Tables

Table 5-1 identifies the BillerXpert tables in the Oracle database.

Table 5-1 Oracle Database Billing Tables

Table Name	Functional Group	Description
1. BXBillerCustomerAccount	Document	BillerXpert customer account
2. BXContainer	Document	Containers
3. BXDocument	Document	Documents
4. BXSummary	Document	Summary lines
5. BXEvent	Events	Events
6. BXEventHandler	Events	Event handlers
7. BXEventQueue	Events	Event queue entries
8. BXHandler	Events	Handlers
9. BXEventLog	Activity Tracking	Event logs
10. BXEventLogDetail	Activity Tracking	Event log detail lines
11. BXEventSchedule	Activity Tracking	Schedule of Events
12. BXActivity	Activity Tracking	Activity logs
13. BXLoadTrack	Other	Track file loading.
14. BXMessage	Other	E-mail message information
15. BXError	Other	Error information
16. BXExtensionDesc	Other	Extension information
17. BXPayHistory	Payment	Payment histories
18. BXPaySchedule	Payment	Payment schedules
19. BXUserAcctAsgt	Assignment tables	Relationship between users and accounts
20. BXDocLineAsgt	Assignment tables	Relationship between document and line item details

Event Tables

This section describes the BillerXpert event tables.

BXEvent Table

The BXEvent table stores all BillerXpert event names. Table 5-2 lists this information.

Table 5-2 BXEvent Table

Name	Required	Type	Description
1. EventID	Yes	number(15)	Primary key — system generated.
2. EventName	Yes	varchar2(64)	Name of the event.
3. Description	No	varchar2(256)	Description of the event.
4. BillerID	Yes	varchar2(32)	Biller ID
5. Status	Yes	varchar2(32)	Status of event. Valid values are: • ACTIVE • INACTIVE • VOID
6. Extension1	No	varchar2(256)	Extension1 for events
7. Extension2	No	varchar2(256)	Extension2 for events
8. Extension3	No	varchar2(256)	Extension3 for events
9. Extension4	No	varchar2(256)	Extension4 for events

BXHandler Tables

The BXHandler table stores event handler information. Table 5-3 lists these handlers.

Table 5-3 BXHandler Table *(1 of 2)*

Name	Required	Type	Description
HandlerID	Yes	number(15)	Primary key — system generated.
HandlerName	Yes	varchar2(64)	Name of the event handler
ApplicationID	No	varchar2(64)	AppLogic name or GUI ID for the event handler.
Description	No	varchar2(256)	Description of the event handler.

Table 5-3 BXHandler Table (2 of 2)

Name	Required	Type	Description
Package	No	varchar2(256)	Used with the classpath field, represents the location of the class responsible for collecting configuration information for the handler. For example, for an email handler, these fields would collect address and subject information.
ClassPath	No	varchar2(256)	Used with the package field, represents the location of the class responsible for collecting configuration information for the handler. For example, for an e-mailhandler, these fields would collect address and subject information.
Extension1	No	varchar2(256)	Extension1 for handlers
Extension2	No	varchar2(256)	Extension2 for handlers
Extension3	No	varchar2(256)	Extension3 for handlers
Extension4	No	varchar2(256)	Extension4 for handlers

BXEventHandler Table

The BXEventHandler table stores the mapping information between events and event handlers. Table 5-4 lists this information.

Table 5-4 BXEventHandler Table (1 of 2)

Name	Required	Type	Description
EventHandlerID	Yes	number(15)	Primary key — system generated.
EventID	Yes	number(15)	Event ID — system generated
Status	Yes	varchar2(32)	Status of event to event handler mapping. Valid values are: - ACTIVE - INACTIVE - VOID
HandlerID	Yes	number(15)	Handler ID — system generated
Extension1	No	varchar2(256)	Extension1 for event handlers
Extension2	No	varchar2(256)	Extension2 for event handlers
Extension3	No	varchar2(256)	Extension3 for event handlers

Table 5-4 BXEventHandler Table (2 of 2)

Name	Required	Type	Description
Extension4	No	varchar2(256)	Extension4 for event handlers

BXEventQueue Table

The BXEventQueue table stores the event data. Table 5-5 lists this information.

Table 5-5 BXEventQueue Table (1 of 2)

Name	Required	Type	Description
EventQueueID	Yes	number(15)	Primary key — system generated.
BillerID	No	varchar2(64)	Biller ID.
EventID	Yes	number (15)	Event ID
Priority	No	number(15)	Not used.
Data	No	varchar2(2000)	Input data for the event. For example, in the case of an email event, where the event sends email to a user, the data might be: name=JMullen@netscape.co m&BX_Template=/tmp/ mail/template
Status	No	varchar2(32)	Status of the record. Values can either be “null” or “PROCESSING.” If a record is being processed, it cannot be processed by any other Event Server.
Pushed	No	date	Date when the event was pushed from the queue.
Popped	No	date	Date when the event was popped from the queue.
Processed	No	date	Date when the event was processed.
EventDate	Yes	date	Date when event is scheduled
EventScheduleID	No	number(15)	Event Schedule ID
Extension1	No	varchar2(256)	Extension1 for event queues
Extension2	No	varchar2(256)	Extension2 for event queues
Extension3	No	varchar2(256)	Extension3 for event queues

Table 5-5 BXEventQueue Table (2 of 2)

Name	Required	Type	Description
Extension4	No	varchar2(256)	Extension4 for event queues

Activity Tracking Tables

The activity tracking tables store activity tracking data.

BXEventLog Table

The BXEventLog table stores event log information. Table 5-6 lists this information.

Table 5-6 BXEventLog Table

Name	Required	Type	Description
EventLogID	Yes	number(15)	Primary key — system generated.
FileID	Yes	number(15)	File ID used by the loader.
CreateDate	yes	date	Date the event was create.
ApplName	Yes	varchar2(32)	Name of the AppLogic that initiated the activity.
Subject	Yes	varchar2(32)	Used with the Action field, identifies the activity associated with the event
Action	Yes	varchar2(32)	Used with the Subject field, identifies the activity associated with the event.
OrgID	No	varchar(64)	Biller ID.
AcctID	No	varchar(64)	Customer account ID.
CreatedBy	No	varchar2(64)	User name.
Message	No	varchar2(512)	Message associated with the activity.
Extension1	No	varchar2(256)	Extension1 for event queues
Extension2	No	varchar2(256)	Extension2 for event queues
Extension3	No	varchar2(256)	Extension3 for event queues
Extension4	No	varchar2(256)	Extension4 for event queues

BXEventLogDetail Table

The BXEventLog Detail Table table stores additional event activity tracking information that is not stored in the BXEventLog. Table 5-7 lists this information.

Table 5-7 BXEventLogDetail Table

Name	Required	Type	Description
EventLogDetailID	Yes	number(15)	Primary key — system generated.
EventLogID	Yes	number(15)	Primary key of BXEventLog table.
FileID	Yes	number(15)	Primary key of BXEventLog table.
Name	Yes	varchar2(32)	Used for additional information not included in the BX_Event_Log table.
Value	No	varchar2(256)	Used for additional information not included in the BX_Event_Log table.
Extension1	No	varchar2(256)	Extension1 for event log detail
Extension2	No	varchar2(256)	Extension2 for event log detail
Extension3	No	varchar2(256)	Extension3 for event log detail
Extension4	No	varchar2(256)	Extension4 for event log detail

BXEventSchedule Table

The BXEventSchedule table stores event scheduling activity information. Table 5-8 lists this information.

Table 5-8 BXEventSchedule Table (1 of 2)

Name	Required	Type	Description
EventScheduleID	Yes	number(15)	Primary key — system generated.
Description	No	varchar2(64)	Description of the event schedule
StartOnDate	Yes	date	Start date of event
StopOnDate	No	date	Stop date of event
ScheduleName	Yes	varchar2(64)	Name of scheduled event
SchMode	Yes	number(15)	Mode of scheduled event
DayOfMonth	No	number(15)	Scheduled day of the month event

Table 5-8 BXEventSchedule Table (2 of 2)

Name	Required	Type	Description
MonthFrequency	No	number(15)	Monthly frequency of the event
Mon	No	number(15)	Day of the week
Tue	No	number(15)	Day of the week
Wed	No	number(15)	Day of the week
Thu	No	number(15)	Day of the week
Fri	No	number(15)	Day of the week
Sat	No	number(15)	Day of the week
Sun	No	number(15)	Day of the week
LastProcessdDate	No	date	Date of last processed event
Extension1	No	varchar2(256)	Extension1 for event log detail
Extension2	No	varchar2(256)	Extension2 for event log detail
Extension3	No	varchar2(256)	Extension3 for event log detail
Extension4	No	varchar2(256)	Extension4 for event log detail

BXActivity Table

The BXActivity table stores event activity information. Table 5-9 lists this information.

Table 5-9 BXActivity Table

Name	Required	Type	Description
ActivityID	Yes	number(15)	Primary key — system generated.
ActivityTime	Yes	date	Time date of activity
UserID	Yes	varchar(128)	User ID of participant in activity
BillerID	Yes	varchar2(128)	Biller ID of activity
CustomerID	No	number(15)	Customer ID of activity
Activity	Yes	varchar(30)	Activity information
ActivityData	No	varchar2(2000)	Activity Data
EmulationID	No	varchar2(128)	Emulation ID
ActivityDetail	Yes	varchar2(2000)	Detail of activity

Document Tables

This section describes the BillerXpert document tables.

BXBillerCustomerAccount Table

The BXBillerCustomerAccount table stores the biller and customer account information. A single customer may have multiple accounts in this table. Biller / Customer will have a unique AcctID for each account in the BillerXpert database. The AcctID is used in conjunction with charge events, payment schedules, and log entries.

Table 5-10 lists this information.

Table 5-10 BXBillerCustomerAccount Table (1 of 3)

Name	Required	Type	Description
AcctID	Yes	number(15)	Customer account ID.
BillerID	Yes	varchar2(128)	Biller ID.
AccountType	No	varchar2(32)	Customer defined account type.
CustomerType	No	varchar2(32)	Customer type. The customer type can be: • REGULAR • SPECIAL
Description	No	varchar2(256)	Brief description of the Customer / Biller account. For example, "Account for residential address."
TaxID	No	varchar2(64)	Biller / Customer tax identification number.
Since	No	number(8)	Date when the customer registered with the site.
ServiceOptions	No	varchar2(1024)	Available service options
Profile	No	varchar2(1024)	This tag represents the Customer / Biller profile information. For example: • AGE • PROFESSION • INCOME • SPECIAL INTERESTS
AccountNumber	Yes	varchar2(32)	Customer account number.

Table 5-10 BXBillerCustomerAccount Table (2 of 3)

Name	Required	Type	Description
AccountStatus	Yes	varchar2(32)	Status of an account. Valid statuses are: • ACTIVE • INACTIVE • PENDING • VOID
UnpaidBills	Yes	number(8)	Total number of bills left unpaid since the last bill date.
UnpaidBalance	Yes	decimal(16,4)	Total unpaid balance. This is a cumulative amount.
AccountBalance	Yes	decimal(16,4)	Current account balance.
AccountBalanceDate	Yes	date	Date when an account balance is reported.
CreateDate	Yes	date	Date on which the customer account was created.
ModifyDate	Yes	date	Date on which the customer account was modified.
DisputeStatus	Yes	varchar2(32)	Dispute status that can indicate any of the following: • If a dispute exists • If a dispute has been acted upon • If a dispute is closed Default values are: • OPEN • CLOSED • REOPEN • NODISPUTE
Extension	No	varchar2(2000)	Extra descriptive information related to the account.
BillingAddress	No	varchar2(64)	Customer's billing address.
ServiceAddress	No	varchar2(64)	Location of the service. This may also be the billing address.
VerifiedBy	No	varchar2(64)	Site implemented: name of the person who verifies account data entries.

Table 5-10 BXBillerCustomerAccount Table (3 of 3)

Name	Required	Type	Description
VerifiedDate	No	date	Site implemented: date on which the account data was verified.
AuthorizedBy	No	varchar2(64)	Site implemented: name of the authorizing person.
AuthorizedDate	No	date	Site implemented: date of authorization.
FileID	Yes	number(15)	File ID generated by the loader when it loads a file into the system.
AutoPayStatus	Yes	varchar(1)	Single character that contains pay status
Extension1	No	varchar2(256)	Extension1 for customer account detail
Extension2	No	varchar2(256)	Extension2 for customer account detail
Extension3	No	varchar2(256)	Extension3 for customer account detail
Extension4	No	varchar2(256)	Extension4 for customer account detail

BXContainer Table

This is the BillerXpert documentation container table. A container has one associated summary but can contain more than one document. Each document has its own associated summary.

Table 5-11 lists these containers.

Table 5-11 BXContainer Table (1 of 2)

Name	Required	Type	Description
ContainerID	Yes	number(15)	Primary key — system generated.
AcctID	Yes	number(15)	Primary key — system generated.
FileID	Yes	number(15)	Primary key — system generated
Description	No	varchar2(256)	Descriptive information about the container.
Header	No	varchar2(128) H	Heading information related to the container.
Footer	No	varchar2(128)	Footer information related to the container.
Extension	No	varchar2(2000)	Additional descriptive information related to the container.

Table 5-11 BXContainer Table (2 of 2)

Name	Required	Type	Description
Extension1	No	varchar2(256)	Extension1 for document detail
Extension2	No	varchar2(256)	Extension2 for document detail
Extension3	No	varchar2(256)	Extension3 for document detail
Extension4	No	varchar2(256)	Extension4 for document detail

BXDocument Table

The BXDocument table is the billing document table. A document has one summary and may consist of many line items. Each document might be associated with a payment history.

Table 5-12 lists the billing document information.

Table 5-12 BXDocument Table (1 of 2)

Name	Required	Type	Description
DocID	Yes	number(15)	Primary key — system generated.
ContainerID	Yes	number(15)	Primary key — system generated.
AcctID	Yes	number(15)	Primary key — system generated.
FileID	Yes	number(15)	Primary key — system generated.
DocType	Yes	varchar2(32)	Document type. Valid document types are: • BILL • INVOICE • NOTICE • STATEMENT
BillType	Yes	varchar2(32)	Billing type. Valid billing types are: • TELCO • CABLE • UTILITY • BANKSTMT • MORTGAGESTMT
CreateDate	Yes	date	Document creation date.
ModifyDate	Yes	date	Document modification date.

Table 5-12 BXDocument Table (2 of 2)

Name	Required	Type	Description
FirstViewDate	No	date	Date on which the document was first viewed.
LastViewDate	No	date	Date on which the document was last viewed.
ViewCount	Yes	number(15)	Number of times the document was viewed. This is the difference between the FirstViewDate and the LastViewDate values.
Description	No	varchar2(256)	Description of the document.
BillDate	Yes	date	Date the bill was generated.
Status	Yes	varchar2(32)	Document status. Valid document statuses are: • ACTIVE • INACTIVE • VOID
CopyType	Yes	varchar2(64)	Document copy type. Valid document copy types are: • ORIGINAL • DUPLICATE
Extension	No	varchar(2000)	Additional descriptive information related to the document.
Extension1	No	varchar2(256)	Extension1 for document detail
Extension2	No	varchar2(256)	Extension2 for document detail
Extension3	No	varchar2(256)	Extension3 for document detail
Extension4	No	varchar2(256)	Extension4 for document detail

BXSummary Table

The BXSummary table is the billing summary table. It contains summary information about the customer’s bill. Table 5-13 lists this information.

Table 5-13 BXSummary Table (1 of 4)

Name	Required	Type	Description
SumID	Yes	number(15)	Primary key — system generated.

Table 5-13 BXSummary Table (2 of 4)

Name	Required	Type	Description
ContainerID	Yes	number(15)	Primary key — system generated.
DocID	Yes	number(15)	Primary key — system generated.
FileID	Yes	number(15)	Primary key — system generated.
StartDate	Yes	date	Start of billing cycle date.
EndDate	Yes	date	End of the billing cycle date.
OriginalAmount	Yes	decimal(16,4)	Original amount of the bill. For example, a bill may be adjusted by a payment or other reason. The original amount is the amount before any of these adjustments.
CurrentBalance	Yes	decimal(16,4)	Current account balance.
CurrentCharge	Yes	decimal(16,4)	Current statement balance.
PreviousBalance	Yes	decimal(16,4)	Previous balance amount.
PreviousPayment	Yes	decimal(16,4)	Previous month's payment amount.
BalanceDate	Yes	date	Statement date.
DueDate	Yes	date	Statement due date.
LateAfterDate	No	date	Date after which the statement is considered late (that is, after the grace period).
PenaltyType	No	varchar2(64)	Penalty type. For example: • FIXED • PERCENT
PenaltyRate	No	decimal(16,4)	Penalty rate. For example: • If fixed, the penalty amount. • If ratio, the penalty percentage.
MinPayAmount	Yes	decimal(16,4)	Minimum payment amount.
PaymentAmount	Yes	decimal(16,4)	Previous payment amount received.
LastPayDate	No	date	Date the last payment was made.
PayStatus	Yes	varchar2(32)	Payment status. For example: • NOT_PAID • FULLY_PAID • PARTIALLY_PAID • OVER_PAID

Table 5-13 BXSummary Table (3 of 4)

Name	Required	Type	Description
DisputeAmount	Yes	decimal(16,4)	Amount of a bill or line item disputed by the customer.
DisputeCaseNum	No	varchar2(64)	Dispute identification number.
DisputeReasonCode	No v	archar2(64)	Dispute reason code. For example: • ALREADY_PAID • INVALID_ITEM • WRONGBILL • WRONGAMOUNT • WRONGITEM • OTHER
DisputeStatus	Yes	varchar2(32)	Dispute status. For example: • OPEN • CLOSED • REOPEN • NODISPUTE
Status	Yes	varchar2(32)	Billing summary status. For example: • ACTIVE • INACTIVE • VOID
PayTermsCode	No	varchar2(32)	For future use. This field can be used for passing along payment terms information from the legacy system.
DiscountBeforeDate	No	date	Date before which a discount will apply.
DiscountType	No	varchar2(32)	Discount type. For example: • FIXED • PERCENT
DicountRate	No	decimal(16,4)	Discount rate. For example: • If fixed, the amount of the discount. • If percent, the percentage of the discount.
Extension	No	varchar2(2000)	Additional descriptive information related to the billing summary.
URL	No	varchar2(256)	Site implemented: stores a URL, for example, to the detail bill.
Extension1	No	varchar2(256)	Extension1 for summary detail

Table 5-13 BXSummary Table (4 of 4)

Name	Required	Type	Description
Extension2	No	varchar2(256)	Extension2 for summary detail
Extension3	No	varchar2(256)	Extension3 for summary detail
Extension4	No	varchar2(256)	Extension4 for summary detail

Payment Related Tables

The Payment Related tables contain information about a customer's payment history and store information each time a customer pays a bill.

BXPayHistory Table

The BXPayHistory table contains a customer's payment history. Table 5-14 lists this information.

Table 5-14 BXPayHistory Table (1 of 3)

Name	Required	Type	Description
PayHistID	Yes	number(15)	Primary key — system generated.
AcctID	Yes	number(15)	Primary key — system generated.
DocID	Yes	number(15)	Primary key — system generated.
Type	Yes	varchar2(32)	Payment type. Valid values are credit card or check payment.
Brand	Yes	varchar2(32)	For credit cards: the type of credit card, for example, VISA.
PmtAmount	Yes	decimal(16,4)	Amount of payment.
CreateDate	Yes	date	Date the payment record was created in the system.
ModifyDate	Yes	date	Date the payment record was modified in the system.

Table 5-14 BXPAYHistory Table (2 of 3)

Name	Required	Type	Description
PmtTxnStatus	Yes	varchar2(32)	Payment transaction status. Valid values are: • AUTH_PENDING • AUTH_SUCCESS • AUTH_FAIL • CAPTURE_PENDING • CAPTURE_IN_PROCESS • CAPTURE_SUCCESS • CAPTURE_FAIL • VOID_PENDING • VOID_SUCCESS • VOID_FAIL • REFUND_PENDING • REFUND_FAIL • NSF • SUBMITTED • CANCELLED • CHECK_SUB_PENDING • CHCK_SUB_IN_PROCESS • CHECK_SUB_SUCCESS • CHECK_SUB_FAILED • CHECK_VENDOR_FAILED • CHECK_VENDOR_PENDING • CHECK_VENDOR_SUCCESS • PENDING
PmtTxnID	Yes	number(15)	Transmission ID associated with the payment.
PmtMode	Yes	varchar2(32)	Payment mode. Valid values are: • AUTO • MANUAL
FileID	Yes	number(15)	File ID used by the loader.
SchedulePmtDate	Yes	date	Date the payment is scheduled.
NumProcessDays	No	number(15)	Number of process days for payment
ServerTrnID1	No	varchar2(1000)	Server Transaction ID1
ServerTrnID2	No	varchar2(1000)	Server Transaction ID2
PmtProcessDate	No	date	Date of payment processing.
Extension	No	varchar2(2000)	Used to store additional user-defined information.
Extension1	No	varchar2(256)	Extension1 for pay history detail

Table 5-14 BXPayHistory Table (3 of 3)

Name	Required	Type	Description
Extension2	No	varchar2(256)	Extension2 for pay history detail
Extension3	No	varchar2(256)	Extension3 for pay history detail
Extension4	No	varchar2(256)	Extension4 for pay history detail

BXPaySchedule Table

The BXPaySchedule table contains information related to automatic payment. This feature can be implemented by the site.

Table 5-15 lists this information.

Table 5-15 BXPaySchedule Table (1 of 2)

Name	Required	Type	Description
PaySchedID	Yes	number(15)	Primary key — system generated.
AcctID	Yes	number(15)	Primary key — system generated.
BillerID	Yes	varchar2(128)	Biller ID
BillerAcctNum	Yes	varchar2(128)	Biller Account Number
StartOnDate	No	date	Date on which the automatic payment starts.
StopOnDate	No	date	Date on which the automatic payment stops.
Frequency	No	varchar(32)	Frequency at which payments are automatically made.
PayMode	Yes	varchar(32)	Mode of payment . Values can be: • ONDUEDATE • BEFOREDUEDATE • ONFIXEDATE
DayBeforeDue	Yes	number(15)	Specifies n number of days to process the payment before the actual due date.
DayOfMonth	Yes	number(15)	Specifies day of the month payments
PayDate	No	date	Date of payment
PayAmount	No	decimal(16,4)	Amount of the payment.
PayAmountThreshold	No	decimal(16,4)	Amount threshold of the payment.

Table 5-15 BXPaySchedule Table (2 of 2)

Name	Required	Type	Description
PayAmountType	Yes	varchar2(64)	Payment amount type.Values can be: • FULL • FIXEDAMOUNT • MINIMUM • FIXEDPERCENT
PmtAcctID	Yes	varchar2(128)	Identifies the account from which to make a payment.
Status	Yes	varchar2(32)	Status. Valid values are: • ACTIVE • INACTIVE • VOID
Extension1	No	varchar2(256)	Extension1 for pay scheduling detail
Extension2	No	varchar2(256)	Extension2 for pay scheduling detail
Extension3	No	varchar2(256)	Extension3 for pay scheduling detail
Extension4	No	varchar2(256)	Extension4 for pay scheduling detail

Assignment Tables

This section describes the assignment tables.

BXUserAcctAsgt Table

The BXUserAcctAsgt table links the customer profile to the billing account. Table 5-16 lists this information.

Table 5-16 BXUserAcctAsgt Table

Name	Required	Type	Description
AcctID	Yes	number(15)	Biller Customer Account (BCA) ID.
UserID	Yes	varchar2(128)	Online user account login name.
Extension1	No	varchar2(256)	Extension1 for pay scheduling detail

Table 5-16 BXUserAcctAsgt Table

Name	Required	Type	Description
Extension2	No	varchar2(256)	Extension2 for pay scheduling detail
Extension3	No	varchar2(256)	Extension3 for pay scheduling detail
Extension4	No	varchar2(256)	Extension4 for pay scheduling detail

BXDocLineAsgt Table

The BXDocLineAsgt table links the document to the line item details. Table 5-17 lists this information.

Table 5-17 BXDocLineAsgt Table

Name	Required	Type	Description
DocID	Yes	number(15)	Primary key — system generated.
LineID	Yes	number(15)	Primary key — system generated.
FileID	Yes	number(15)	Primary key — system generated.
LineType	Yes	varchar2(32)	Interface to the detail table.
TableName	Yes	varchar2(32)	Line item's table name.
Extension1	No	varchar2(256)	Extension1 for line detail.
Extension2	No	varchar2(256)	Extension2 for line detail
Extension3	No	varchar2(256)	Extension3 for line detail.
Extension4	No	varchar2(256)	Extension4 for line detail.

Other Tables

This section lists other database tables that were not discussed previously.

BXLoadTrack Table

This table is used by the loading during the file loading process. Table 5-18 lists this information.

Table 5-18 BXLoadTrack Table

Name	Required	Type	Description
FileID	Yes	number(15)	Primary key — system generated.
BillerID	Yes	varchar2(128)	Biller ID.
Status	Yes	varchar2(16)	Status of the load procedure.
Path	Yes	varchar2(512)	Loader file path.
RecordCount	No	number(20)	Number of records inserted
FileType	No	varchar2(10)	Type of file inserted in the system.
LoadDate	No	date	Loading date
Extension1	No	varchar2(256)	Extension1 for load detail
Extension2	No	varchar2(256)	Extension2 for load detail
Extension3	No	varchar2(256)	Extension3 for load detail
Extension4	No	varchar2(256)	Extension4 for load detail

BXMessage Table

The BXMessage table is used to store email message information. Table 5-19 lists this information.

Table 5-19 BXMessage Table (1 of 2)

Name	Required	Type	Description
MessageID	Yes	number(15)	Primary key — system generated.
BillerID	Yes	varchar2(128)	Primary key — system generated.
AcctID	Yes	number(15)	Primary key — system generated.
DocID	Yes	number(15)	Primary key — system generated.
CreateDate	Yes	date	Date the message was created.
MsgTo	Yes	varchar2(32)	Destination address of the message.

Table 5-19 BXMessage Table (2 of 2)

Name	Required	Type	Description
MsgFrom	Yes	varchar2(32)	Sending address of the message.
Subject	Yes	varchar2(64)	Subject of the message.
Type	Yes	varchar2(32)	Type of message. Valid values are: • CORRESPONDENCE • DISPUTE
Message	Yes	varchar2(2000)	Content of the message.
SenderType	Yes	varchar2(32)	Sender type. Valid values are: • CUSTOMER • ADMIN • CSR
CaseNum	No	varchar2(32)	Case number for customer support reference.
CsrName	No	varchar2(64)	CSR name.
Status	Yes	varchar2(32)	Status. Valid values are: • ACTIVE • INACTIVE • VOID
Extension	No	varchar2(2000)	Used to store additional user-defined information.
Extension1	No	varchar2(256)	Extension1 for message detail
Extension2	No	varchar2(256)	Extension2 for message detail
Extension3	No	varchar2(256)	Extension3 for message detail
Extension4	No	varchar2(256)	Extension4 for message detail

BXExtensionDesc Table

The BXExtensionDesc table stores information about extensions. Table 5-20 lists this information.

Table 5-20 BXExtensionDesc Table

Name	Required	Type	Description
Tablename	Yes	varchar2(256)	Table extension name
Fieldname	Yes	varchar2(256)	Field extension name

Table 5-20 BXExtensionDesc Table

Name	Required	Type	Description
Description	Yes	varchar2(256)	Description extension
Extension1	No	varchar2(256)	Extension1 for detail
Extension2	No	varchar2(256)	Extension2 for detail
Extension3	No	varchar2(256)	Extension3 for detail
Extension4	No	varchar2(256)	Extension4 for detail

BXError Table

This BXError table is used to store error information. Table 5-21 lists this information.

Table 5-21 BXError Table

Name	Required	Type	Description
UserID	Yes	varchar2(128)	UserID name
EmulationID	No	varchar2(128)	Emulation ID name
BillerID	Yes	varchar2(128)	BillerID name
ErrorTime	Yes	date	Time of error
ErrorCode	Yes	varchar2(30)	Code error
ErrorLevel	No	number(10)	Error level
ErrorDetail	No	varchar2(2000)	Error detail information

Membership Schema

This chapter describes the structure of the LDAP Directory Server database used by BillerXpert.

This chapter includes the following sections:

- Membership Schema
- Relationship Between Tables
- Listing of Structures
- Biller Tables
- Customer Tables

CAUTION BillerXpert provides an API that you should use to access the database instead of the schema. The database schema for BillerXpert is subject to change in future versions of BillerXpert. If you rely on the schema, you should consider the potential reimplementing effort that you could incur as the result of an upgrade to the database.

Membership Schema

The database schema for membership consists of tables that represent the following kinds of information:

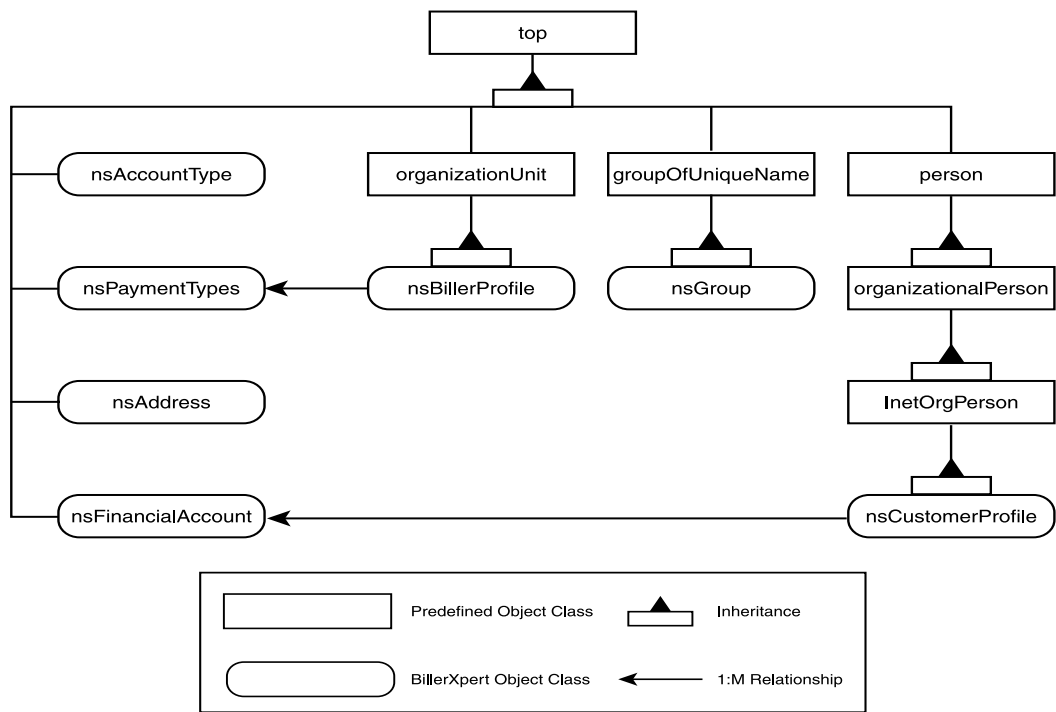
- Biller information
- Financial account information global for the biller site
- Customer information

- Customer financial account information
- Customer payment information

Relationship Between Tables

BillerXpert uses fields from some Directory Server tables, which are also listed in the following membership tables. Figure 6-1 shows the relationships of these tables.

Figure 6-1 Membership Table Relationships



Listing of Structures

Table 6-1 lists the BillerXpert Oracle database tables.

Table 6-1 Membership Tables

Table Name	Functional	Grouping Description
nsAccountType	Biller	Stores financial account information. Global for the biller site.
nsPaymentType	Biller	Stores biller payment information.
nsAddressTable	Customer	Stores addresses of customers.
nsBillerProfile	Biller	Stores biller information.
nsCustomerProfile	Customer	Stores customer information.
nsFinancialAccount	Customer	Stores information about the customer's financial account.
nsGroup	Other	Stores group information for the biller site.

The field type in the membership tables can be one of the following:

- CES (Case Exact String — or case sensitive string)
- CIS (Case Ignore String — or not case sensitive string)
- Telephone

Biller Tables

The Biller tables store information related specifically to the biller or to the biller site.

nsBillerProfile Table

The nsBillerProfile table stores information about the biller. Table 6-2 lists this information.

Table 6-2 nsBillerProfile Table (1 of 3)

Name	Required	Type	Description
OU	Y	CIS string	Organization unit

Table 6-2 nsBillerProfile Table (2 of 3)

Name	Required	Type	Description
nsName	N	CES string	Biller name.
nsStatus	N	CIS	string Biller status.
nsType	N	CIS string	Biller organization type, for example, Direct.
nsEmail	N	CIS string	Biller email.
nsURL	N	CIS string	Biller URL.
nsChartset	N	CIS string	Character set.
nsLanguage	N	CIS string	Language set
nsCreateddate	N	CIS string	Date the biller was created.
nsCreatedby	N	CIS string	Name of the person who created the biller.
nsCurrency	N	CIS string Currency type.	The default is US dollars.
nsLastModifies	N	CIS string	Name of person who last modified the biller information.
mail	N	CIS string	Biller email address.
nsPayeeID	N	CIS string	Biller payee ID.
nsPrimarycontact	N	CIS string	Biller primary contact name.
nsPrimarycontactPhone	N	CIS string	Biller primary contact phone.
nsPrimarycontactEmail	N	CIS string	Biller primary contact email
nsProfileextension	N	CIS string	Biller profile extension. Used to store additional biller profile information.
nsSecondarycontact	N	CIS string	Biller secondary contact name
nsSecondarycontactPhone	N	CIS string	Biller secondary contact phone
nsSecondarycontactEmail	N	CIS string	Biller secondary contact email
nsBillerCycleDate	N	CIS string	Biller cycle date
nsBillerBillMonth	N	CIS string	Biller bill month
nsCustEnrollOption	N	CIS string	Biller custom enroll option
nsCustBillPresentment	N	CIS string	Biller bill presentment
nsCustAutoPay	N	CIS string	Biller custom auto pay

Table 6-2 nsBillerProfile Table (3 of 3)

Name	Required	Type	Description
nsServiceextension	N	CIS string	Biller service extension. Used to store biller defined service options.
nsTimezone	N	CIS string	Biller's time zone
nsAddress1	N	CIS string	Biller's address
nsAddress2	N	CIS string	Biller's address
nsAddress3	N	CIS string	Biller's address 3
nsCity	N	CIS string	Biller's city
nsCountry	N	CIS string	Biller's country
nsTaxid	N	CIS	string Biller's tax identification
nsMerchantid	N	CIS string	Biller's merchant identification
nsgroup	N	CIS string	Biller group
nsInstantGratification	N	CIS string	Biller Instant Gratification
nsInstantGratificationURL	N	CIS string	Biller Instant Gratification URL
nsExtension	N	CIS string	Biller extension

nsAccountType Table

The nsAccountType table stores account type information for a biller. This information is global for the domain.

Table 6-3 lists this information.

Table 6-3 nsAccountType Table (1 of 2)

Name	Required	Type	Description
nsGuid	Y	CIS string (primary key)	Global unique ID. Primary key.
cn	N	CIS string	common name
description	N	CIS string	Description
nsOwner	N	CIS string	string Owner
nsServiceOption	N	CIS string	Service option
nsStatus	N	CIS string	Account status

Table 6-3 nsAccountType Table (2 of 2)

Name	Required	Type	Description
nsName	N	CES string	Account type name, for example “checking”

nsGroup Table

The nsGroup table stores information about the supported groups. For BillerXpert the supported groups are Administrator and default.

Table 6-4 lists this information.

Table 6-4 nsGroup Table

Name	Required	Type	Description
cn	N	CIS string	common name
description	N	CIS string	Description
nsType	N	CIS string	Type
nsacl	N	CIS string	Access control list
nsstatus	N	CIS string	status

nsPaymentType Table

This table stores payment related information for a customer or biller. Multiple instances of this information can exist for a single customer.

Table 6-5 lists this information.

Table 6-5 nsPayment Table (1 of 2)

Name	Required	Type	Description
nsGuid	Y	CIS string (primary key)	Global unique ID. Primary key.
nstype	N	CIS string	Payment type, for example, credit card or check.
nsBrand	N	CIS string	Brand name of the credit card, for example, VISA or Diners.
description	N	CIS string	Description
nsMaxAmount	N	CIS string	Maximum amount

Table 6-5 nsPayment Table (2 of 2)

Name	Required	Type	Description
nsminAmount	N	CIS string	Minimum amount
nsName	N	CES string	Payment name, for example, MASTER (short name for a credit card).
nsOwner	N	CIS string	Name of the person responsible for the payment.
nsStatus	N	CIS string	Payment status. These values can be Active or Inactive. Inactive disables the payment method for the customer.

Customer Tables

The customer tables store information on a customer. Some of this information can exist in multiple instances for a single customer. For example, a customer may have more than one financial account from which he pays bills. In this case, the customer will have two or more instances of financial account data.

nsCustomerProfile Table

The nsCustomer table stores information about a customer. Table 6-6 lists this information.

Table 6-6 nsCustomerProfile Table (1 of 3)

Name	Required	Type	Description
UID	Y	CIS string	User ID
nsAccount	N	CIS string	Account ID
nsStatus	N	CIS string	Customer status. The values are Active or Inactive.
nsType	N	CIS string	Customer type. This information is biller defined.
nsUrl	N	CIS string	Customer URL.
nsCharset	N	CIS string	Character set.
nsCreateddate	N	CIS string	Date the customer was created.

Table 6-6 nsCustomerProfile Table (2 of 3)

Name	Required	Type	Description
nsCreatedby	N	CIS string	Name of person who created the customer.
nsCurrency	N	CIS string	Currency type.
nsLastModified	N	CIS string	Name of person who last modified the customer information.
nsMiddleinitial	N	CIS string	Middle initial of the customer
nsNickName	N	CIS string	Customer nickname.
nsPasswordLastChanged	N	CIS string	Date the password was last changed.
nsPrefix	N	CIS string	Customer prefix, for example, Mr. or Miss.
mail	N	CIS string	Customer email address.
nsPrimarycontact	N	CIS string	Customer primary contact name.
nsProfileextension	N	CIS string	Customer profile extension. The product provides predefined profiles and the biller can define its own.
nsSecondarycontact	N	CIS string	Customer secondary contact name.
nsSecretWord	N	CIS string	Customer secret word. This can be used for additional security purposes. For example, the customer's mother's maiden name.
nsServiceextension	N	CIS string	Customer service extension. These are biller defined.
nsTimezone	N	CIS string	Customer time zone.
nsSuffix	N	CIS string	Customer suffix.
nsAddressID (nsAddress)	N	CIS string	Customer address ID.
nspasswordhint	N	CIS string	Customer password hint.
nsgroup (nsGroup)	N	CIS string	Customer group.
nsAccountType (nsAccountType)	N	CIS string	Customer account type ID.

Table 6-6 nsCustomerProfile Table (3 of 3)

Name	Required	Type	Description
nsDefaultPmtacct (nsFinancialAccount)	N	CIS string	Customer default payment account ID.
nsacl	N	CIS string	Access control list.
nsExtension	N	CIS string	Customer extension. Biller defined.
nsFileID	N	CIS string	Customer file ID
nsMaidenName	N	CIS string	Customer's maiden name
nsBirthDate	N	CIS string	Customer's birthdate
nsTaxid	N	CIS string	Customer's tax id
nsdefaultpmtacct	N	CIS string	Customer's default payment account

nsFinancialAccount Table

This table stores information about the customer's financial account. Multiple instances can exist for a single customer.

Table 6-7 lists this information.

Table 6-7 nsFinancialAccount Table (1 of 2)

Name	Required	Type	Description
nsGuid	Y	CIS string (primary key)	Global unique ID. Primary key.
description	N	CIS string	Description
cn	N	CIS string	common name
nsPanhandle	N	CIS string	Pan handle.
nsOwner	N	CIS string	Owner
nsName	N	CIS string	Account name
nsPayhandle (nsPaymentType)	N	CIS string	Pay handle
nsStatus	N	CIS string	Account Status
nsType	N	CIS string	Financial account type
nsExtension	N	CIS string	Extension

Table 6-7 nsFinancialAccount Table (2 of 2)

Name	Required	Type	Description
nsRoutingNumber	N	CIS string	Routing number
nsAccountNumber	N	CIS string	Account number

nsAddress Table

This table stores the customer address information. Multiple instances can exist for a single customer. Table 6-8 lists this information.

Table 6-8 naAddress Table

Name	Required	Type	Description
cn	N	CIS string	common name
nsGuid	Y	CIS string (primary key)	Global unique ID. Primary key.
nsType	N	CIS string	Address type
nsAddress1	N	CIS string	Address 1
nsAddress2	N	CIS string	Address 2
nsAddress3	N	CIS string	Address 3
nsCity	N	CIS string	City name
nsCountry	N	CIS string	Country name
PostalCode	N	CIS string	Postal code
St	N	CIS string	State
nsStatus	N	CIS string	Status

Payment Tables

This chapter describes the structure of the Oracle tables used by BillerXpert to handle payments.

This chapter includes the following topics:

- Payment Schema
- Payment Information Table

CAUTION BillerXpert provides an API that you should use to access the database instead of the schema. The database schema for BillerXpert is subject to change in future versions of BillerXpert. If you rely on the schema, you should consider the potential reimplementation effort that you could incur as the result of an upgrade to the database.

Payment Schema

The payment database table represents payment-related information such as credit card information.

Payment Information Table

This section describes the Oracle database payment information tables.

BXCardInfo

The BXCardInfo table stores credit information. Table 7-1 lists this information.

Table 7-1 BXCardInfo Table

Name	Required	Type	Description
CardInfo	Yes	number(15)	PAN ID for the credit card
Description	No	varchar2(240)	Credit card description
CardType	No	varchar2(40)	Type of credit card such as credit card, debit card, smart card.
CardBrand	No	varchar2(20)	Brand of card such as MASTER, VISA, AMEX, etc.
CardOwner	No	varchar2(40)	Credit card owner
CardNumber	No	varchar2(240)	Credit card number
CardPin	No	varchar2(20)	Credit card personal ID number
CardExpDate	No	varchar2(10)	Credit card expiration date
CardBillName	No	varchar2(40)	Credit card bill name
CardBillStreet	No	varchar2(40)	Credit card bill street address
CardBillCity	No	varchar2(40)	Credit card bill city
CardBillState	No	varchar2(40)	Credit card bill state
CardBillZip	No	varchar2(40)	Credit card bill zip code
CardBillCountry	No	varchar2(3)	Credit card bill country
CardStatus	No	varchar2(20)	Status of card.

Currency Conversion Tables

This chapter describes the structure of the Oracle tables used by BillerXpert to handle currency conversion.

This chapter contains the following topics:

- Currency Conversion Schema
- Listing of Tables
- Currency Conversion Tables

CAUTION BillerXpert provides an API that you should use to access the database instead of the schema. The database schema for BillerXpert is subject to change in future versions of BillerXpert. If you rely on the schema, you should consider the potential reimplement effort that you could incur as the result of an upgrade to the database.

Currency Conversion Schema

The currency conversion database tables represents information about currency data, currency load information, currency conversion results and load tracking status.

Listing of Tables

This section lists the Oracle database currency tables. Table 8-1 lists these tables.

Table 8-1 Currency Tables

Table Name	Functional	Grouping Description
Currency	Currency	Tracks currency code information.
CurrencyLoadTrack	Currency	Tracks currency loading information.
CurrencyConversion	Currency	Tracks currency conversion data.

Currency Conversion Tables

This section describes the currency conversion tables.

Currency Table

The Currency table stores currency code information. Table 8-2 lists this information.

Table 8-2 Currency Table

Name	Required	Type	Description
CurCode	Yes	char(3)	Stores currency acronym
CurID	Yes	number(10)	Stores currency ID
MinorUnit	Yes	number (5)	Maintains minor unit
CurDesc	No	varchar2(240)	Stores currency description
CreateDate	No	date()	Stores date of currency information
EUFlag	No	char(1)	Status bit for Euro flag

CurrencyLoadTrack Table

This table stores currency load tracking information. Table 8-3 lists this information.

Table 8-3 CurrencyLoadTrack Table

Name	Req	Type	Description
FileID	Yes	number(15)	Stores load file ID
FilePath	Yes	varchar2(240)	Stores the path to the load file
RecordCount	Yes	number (10)	Stores the record count
LoadDate	Yes	date()	Stores the date of the data load
Status	Yes	varchar2 (16)	Maintains the status of the load
Source	No	varchar2 (240)	Stores the source of the data feed.
ConvDate	Yes	date()	Stores the currency conversion date.

CurrencyConversion

This table stores currency conversion information. Table 8-4 lists this information.

Table 8-4 CurrencyConversion Table

Name	Req	Type	Description
ConvID	Yes	number(15)	Stores conversion ID
FromCurr	Yes	char (3)	Stores primary currency such as USD
ToCurr	Yes	char (3)	Stores secondary currency
Factor	Yes	number (20, 8)	Stores factor information
FileID	Yes	number (15)	Stores currency file ID

Enumerated Domains

This appendix describes the domains that an implementation of a component business object interface checks before performing actions against a data repository.

This appendix contains the following sections:

- Domain Checking
- The enumdef.conf File

Domain Checking

BillertXpert has a concept of a domain, which restricts some kinds data in data repositories to specific values. A domain is enforced when access to a data repository occurs, not when the data is set. For example, you can call an IBXBillerProfile object's `setBillerStatus()` method and set any value. If a value outside of the domain is detected while executing `IBXEntity.ejbCreate()` method or the `ejbStore()` method, an exception is raised.

The allowed values are determined by the contents of the `enumdef.conf` configuration file. For example, the valid values for status are specified by the following lines:

```
"STATUS" NTV {  
    "value" StrArr [ "ACTIVE", "INACTIVE", "PENDING", "VOID" ],  
    "default" Str "ACTIVE"
```

In this case, the valid values are 'ACTIVE', 'INACTIVE', 'PENDING', and 'VOID'. If a value has not been set, for example, if the `setBillerStatus()` method was not called before calling the `IBXEntity.ejbCreate()` method, the default value, 'ACTIVE' would be used.

The enumdef.conf File

The enumdef.conf file contains the domains that are used by BillerXpert. Each domain, such as STATUS, has associated with it an array of valid values and a default value.

Table A-1 shows the contents of this file.

Table A-1 enumdef.conf File (1 of 5)

<pre>NTV-ASCII { "STATUS" NTV { "value" StrArr ["ACTIVE", "INACTIVE", "PENDING", "VOID"], "default" Str "ACTIVE" }, "CUSTOMERTYPE" NTV { "value" StrArr ["REGULAR", "SPECIAL"], "default" Str "REGULAR" }, "DISPUTESTATUS" NTV { "value" StrArr ["NODISPUTE", "OPEN", "REOPEN", "CLOSED"], "default" Str "NODISPUTE" }, "DISPUTEREASON" NTV { "value" StrArr ["ALREADY_PAID", "INVALID_ITEM", "WRONGAMOUNT", "WRONGITEM", "OTHER", ""], "default" Str "ALREADY_PAID" }, "DOCTYPE" NTV { "value" StrArr ["BILL", "INVOICE", "NOTICE", "STATEMENT"], "default" Str "BILL" }, "BILLTYPE" NTV { "value" StrArr ["TELCO", "CABLE", "UTILITY", "BANKSTMT", "MORTGAGESTMT"], "default" Str "TELCO" }, "COPYTYPE" NTV { "value" StrArr ["ORIGINAL", "DUPLICATE"], "default" Str "ORIGINAL" }, }</pre>

Table A-1 enumdef.conf File (2 of 5)

```

"AUTOPAYFREQUENCY" NTV {
    "value"    StrArr ["PAYWHENDUE" ],
    "default"  Str  "PAYWHENDUE"
},

"PAYMODE" NTV {
    "value"    StrArr ["ONDUEDATE", "BEFOREDUEDATE",
                      "ONFIXEDDATE"],
    "default"  Str  "ONDUEDATE"
},

"PAYAMOUNTTYPE" NTV {
    "value"    StrArr ["FULL", "FIXEDAMOUNT", "MINIMUM",
                      "FIXEDPERCENT"],
    "default"  Str  "FULL"
},

"PENALTYTYPE" NTV {
    "value"    StrArr ["FIXED", "PERCENT" ],
    "default"  Str   "FIXED"
},

"PAYSTATUS" NTV {
    "value"    StrArr ["NOT_PAID", "FULLY_PAID", "PARTIALLY_PAID",
                      "OVER_PAID" ],
    "default"  Str  "NOT_PAID"
},

"DISCOUNTTYPE" NTV {
    "value"    StrArr ["FIXED", "PERCENT" ],
    "default"  Str  "FIXED"
},

"MESSAGETYPE" NTV {
    "value"    StrArr ["CORRESPONDENCE", "DISPUTE" ],
    "default"  Str  "CORRESPONDENCE"
},

"MESSAGESENDERTYPE" NTV {
    "value"    StrArr ["CSR", "ADMIN", "CUSTOMER" ],
    "default"  Str  "CUSTOMER"
},

```

Table A-1 enumdef.conf File (3 of 5)

```

"PMTMODE" NTV {
    "value"    StrArr ["MANUAL","AUTO" ],
    "default" Str    "MANUAL"
},

"BILLERTYPE" NTV {
    "value"    StrArr ["DIRECT","BSP", "CUP" ],
    "default" Str    "DIRECT"
},

"PAYTYPE" NTV {
    "value"    StrArr ["CREDIT CARD","CHECK" ],
    "default" Str    "CREDIT CARD"
},

"CURRENCY" NTV {
    "value"    StrArr ["USD", "CAD", "FRF", "ARS", "ATS", "AUD",
        "BEF", "BRC", "CHF", "CSK", "DEM", "DKK",
        "ESP", "FIM", "GBP", "GRD", "HKD", "INR",
        "IDR", "IEP", "ITL", "JPY", "KRW", "MXP",
        "MYR", "NLG", "NOK", "NZD", "PLZ", "PTE",
        "SEK", "SGD", "ZAR" ],
    "default" Str    "USD"
},

"CARDBRAND" NTV {
    "value"    StrArr ["VISA", "MASTERCARD", "AMEX", "DISCOVER",
        "DINERSCLUB", "CARTEBLANCHE",
        "DDE-CHECKING", "DDE-SAVING"],
    "default" Str    "VISA"
},

"PAYMENTMETHOD" NTV {
    "value"    StrArr ["VISA", "MASTERCARD", "AMEX", "DISCOVER",
        "DINERSCLUB", "CARTEBLANCHE", "CHECK",
        "DDE-CHECKING", "DDE-SAVING"],
    "default" Str    "VISA"
},

DUMMYPAYMENTMETHOD NTV {
    "value"    StrArr ["DUMMY_SM","DUMMY_CC" ],
    "default" Str    "DUMMY_SM"
},

```

Table A-1 enumdef.conf File (4 of 5)

```

"CARDTXNSTATUS" NTV {
    "value"    StrArr ["SUBMITTED", "AUTH_PENDING",
                      "AUTH_SUCCESS", "AUTH_FAIL", "DELETED",
                      "CAPTURE_PENDING", "CAPTURE_IN_PROCESS",
                      "CAPTURE_SUCCESS", "CAPTURE_FAIL",
                      "VOID_PENDING", "VOID_SUCCESS", "VOID_FAIL",
                      "REFUND_PENDING", "REFUND_FAIL", "NSF" ],
    "default" Str    "SUBMITTED"
},

"CHECKTXNSTATUS" NTV {
    "value"    StrArr ["CHECK_SUB_PENDING", "CHECK_SUB_SUCCESS",
                      "CHECK_SUBMITTED", "CHECK_SUB_FAILED",
                      "DELETED", "CHECK_SUB_RETRY",
                      "CHECK_MAX_RETRY", "CHECK_PENDING",
                      "CHECK_SUCCESS", "CHECK_FAILED",
                      "CHECK_NO_HOPE"],
    "default" Str    "CHECK_SUB_PENDING"
},

"TXNSTATUS" NTV {
    "value"    StrArr ["SUBMITTED", "CANCELLED", "DELETED",
                      "AUTH_PENDING", "AUTH_SUCCESS", "AUTH_FAIL",
                      "CAPTURE_PENDING", "CAPTURE_IN_PROCESS",
                      "CAPTURE_SUCCESS", "CAPTURE_FAIL",
                      "VOID_PENDING", "VOID_SUCCESS", "VOID_FAIL",
                      "REFUND_PENDING", "REFUND_FAIL", "NSF",
                      "CHECK_SUB_PENDING", "CHECK_SUB_SUCCESS",
                      "CHECK_SUBMITTED", "CHECK_SUB_RETRY",
                      "CHECK_SUB_FAILED", "CHECK_MAX_RETRY",
                      "CHECK_PENDING", "CHECK_SUCCESS",
                      "CHECK_FAILED", "CHECK_NO_HOPE", "FAILED",
                      "PENDING"],
    "default" Str    "SUBMITTED"
},

"BATCHTYPE" NTV {
    "value"    StrArr ["GENERAL", "SALES", "REFUND" ],
    "default" Str    "GENERAL"
},

```

Table A-1 enumdef.conf File (5 of 5)

<pre>"BATCHSTATUS" NTV { "value" StrArr ["PENDING", "SUCCESS", "PARTIAL_SUCCESS", "MISMATCH_SUCCESS", "DUPLICATE","RETRY", "RESUBMIT","FAILURE", "MAX_SUBMISSIONS", "HALTED", "SUSPENDED", "ADMIN_INIT"], "default" Str "SUCCESS" }, "TXNTYPE" NTV { "value" StrArr ["DO_AUTH", "DO_VOID", "DO_REFUND", "DO_CAPTURE", "CHECK_AUTH"], "default" Str "DO_AUTH" }, "CHECKUSE" NTV { "value" StrArr ["BUSINESS", "PERSONAL"], "default" Str "PERSONAL" }, "PAYMENTVENDOR" NTV { "SM" Str "SiMulator", "CC" Str "CyberCash" }, }</pre>

Index

A

- ACGUIDGeneration class 38
- action constant 58
- activity constants 56
- addCondition method 87, 90
- addOrder method 110
- authenticate method 341

B

- Base and Core Classes 35
- beginTransaction method 369
- BillerXpert documentation set 22
- Billing classes 137
- BXBillerCustomerAccount table 426
- BXBillingContext class 162
- BXBillingContext constructor 163
- BXBillingMgr class 167
- BXContainer table 428
- BXDocLineAsgt table 437
- BXDocument table 429
- BXEvent table 420
- BXEventContext class 389
- BXEventHandler table 421
- BXEventLog table 423
- BXEventLogDetail table 424
- BXEventMgr class 390, 391, 392, 401, 410
- BXEventQueue table 422

BXFilter class 85
BXHandler table 420
BXLoadTrack table 438
BXMembershipMgr class 373
BXMessage table 438
BXOrder class 113
BXPayHistory table 433
BXPaySchedule table 435
BXSummary table 430
BXUserAcctAsgt table 436

C

clear method 116
commitTransaction method 369
count method 117
createAccountType method 374
createBillerCustAcct method 168, 169
createContainer method 169
createCustomerProfile method 375
createDetailMisc method 169
createDetailRating method 169
createEvent method 399
createEventHandler method 400
createEventLog method 170
createEventQueue method 400
createFinancialAccount method 376
CreateGUID method 39
createHandler method 400
createMessage method 170
createObject method 376
createPayHistory method 171
createPaymentType method 376
createPayVendor method 452, 454
createSummary method 171

D

dump method 129

E

ejbAssociateUser method 143, 367
ejbCreate method 129
ejbCreateAccountType method 317
ejbCreatePaymentType method 318
ejbFindAllAccountTypes method 319
ejbFindAllBCAs method 143
ejbFindAllByIdList method 342
ejbFindAllContainers method 144
ejbFindAllDetailsByType method 200
ejbFindAllDocuments method 144, 186
ejbFindAllPaymentHistories method 201
ejbFindAllUserIDs method 145
ejbFindByAccountNumber method 145
ejbFindByContainerID method 285
ejbFindByDocumentID method 285
ejbFindByEventName method 386, 394, 403, 404
ejbFindByHandlerName method 411
ejbFindByLoginName method 343
ejbFindByOrganizationName method 321
ejbFindEventDetails method 217
ejbFindPaymentSchedule method 146
ejbLoad method 131
ejbRemoveAccountType method 322
ejbRemoveAll method 386, 394, 412
ejbStoreAccountType method 323
ejbStoreFinancialAccount method 323, 344
ejbStorePaymentType method 323
ejbXFindAll method 403
enumdef.conf file 458

F

factory classes 25

finalize method 369

G

getAccount method 344
getAccountID method 187, 256, 270, 271
getAccountNumber method 148
getAccountStatus method 148
getAccountType method 148
getAcctID method 217, 241
getAction method 217
getAddress method 345
getAddress1 method 308
getAddress2 method 308
getAddress3 method 308, 309
getAlias method 364, 378
getApplicationName method 412
getBalanceDate method 285
getBillDate method 202
getBillerID method 148, 242, 394, 405
getBillerStatus method 325
getBillingAddress method 148
getBillingContext method 165, 166
getBillType method 202
getBusinessCategory method 326
getBXStatus method 370
getCaseNum method 242
getCharSet method 326, 345
getCity method 309
getContainerID method 203, 286
getContext method 73
getCopyType method 203
getCountry method 309
getCreateDate method 203, 218, 242, 256, 326, 345
getCreatedBy method 218, 326
getCurrency method 327, 346
getCurrentBalance method 286
getCurrentCharge method 286
getCustomerStatus method 346
getCustomerType method 149

getDayPhone method 346
getDescription method 149, 187, 203, 287, 327, 347, 364, 365, 379
getDiscountBeforeDate method 286
getDiscountRate method 287
getDiscountType method 287
getDisputeAmount method 287
getDisputeStatus method 149
getDocID method 242, 257
getDocStatus method 203
getDocType method 204
getEmail method 327, 347
getEndDate method 288
getEventContext method 391
getEventData method 405
getEventDesc method 386
getEventHandler method 395
getEventHandlerStatus method 395
getEventID method 386
getEventName method 387, 395, 396
getEventPopDate method 405
getEventPriority method 405
getEventProcessDate method 406
getEventPushDate method 406
getEventStatus method 406, 407
getExtension method 150, 187, 197, 204, 208, 257, 288, 327, 347
getFaxNumber method 328, 348
getFileID method 204, 209, 257, 288
getFirstName method 348
getFirstViewDate method 204
getFooter method 188
getFrequency method 271
getGroup method 348
getHandlerClassPath method 412
getHandlerDesc method 413
getHandlerID method 413
getHandlerName method 413
getHandlerPackage method 413
getHeader method 188
getLanguage method 328, 348
getLastName method 349
getLastPayDate method 289
getLastViewDate method 205

getLateAfterDate method 289
getLoginName method 349
getMaxAmount method 379
getMembershipContext method 372
getMerchantID method 328
getMessage method 218, 243
getMiddleName method 349
getMinAmount method 379, 380
getMinPayment method 289
getModifyDate method 150, 205, 257, 328, 349
getMsgFrom method 243
getMsgStatus method 243
getMsgTo method 243
getName method 132
getNickName method 349
getNightPhone method 350
GetObjectType method 40
getOrgID method 218
getOrgName method 329, 350
getOriginalAmount method 289
getPanHandle method 365
getPassword method 350
getPasswordLastChanged method 351
getPayAmount method 272
getPayAmountType method 273
getPaymentAmount method 290
getPaymentBrand method 257
getPaymentHandle method 365
getPaymentType method 258
getPayStatus method 290
getPaytermsCode method 290
getPenaltyRate method 290
getPenaltyType method 290
getPhone method 329
getPmtAcctID method 273
getPmtAmount method 258
getPmtMode method 258
getPmtTxnID method 258
getPmtTxnStatus method 259, 260
getPostalCode method 309
getPrefix method 351
getPreviousBalance method 291

getPreviousPayment method 291
getPrimaryContact method 329, 351
getPrimaryKey method 132
getProfile method 150
getProfileExtension method 329, 352
getProperty method 63, 64
getSchedulePmtDate method 259
getScheduleStatus method 273, 277
getSecondaryContact method 330, 352
getSenderType method 244
getServiceAddressID method 150
getServiceExtension method 330, 352
getServiceOptions method 151
getSince method 151
getSize method 110
getStartDate method 291
getStartOnDate method 273
getState method 310
getStatus method 370
getSubject method 219, 221, 244
getSuffix method 353
getSummary method 205
getSumStatus method 292
getTaxID method 151, 330
getTimeZone method 330, 353
getType method 244, 330, 365, 380
getUnpaidBalance method 151
getUnpaidBills method 151
getURL method 292, 331
getUserID method 152
getViewCount method 205

I

IBXAddress class 312
IBXBillerProfile class 312
IBXCardInfo class 452, 454
IBXDetailCall class 198
IBXDocument class 198
IBXEntity class 74, 162

- IBXEvent class 401
- IBXEventEntity class 392
- IBXEventHandler class 392
- IBXEventLog class 215
- IBXEventQueue class 401
- IBXFinancialAccount class 362
- IBXHandler class 410
- IBXMembershipEntity class 371, 377
- IBXMessage class 221
- IBXPayHistory class 254
- IBXPaymentType class 377
- IBXPaySchedule class 268
- IBXSummary class 282
- iPlanet
 - customer support 22

L

- Log method 370

M

- membership classes 29, 306
- miscellaneous constants 115

N

- name constants 114
- nsAddress table 450
- nsBillerProfile table 443
- nsCustomerProfile table 447
- nsFinancialAccount table 449
- nsGroup table 446
- nsPaymentType table 446

O

Open Financial Exchange (OFX) 251
Oracle database tables 31

P

peek method 118
pop method 118
process method 119
push method 119

R

release method 73
result code constants 113
rollbackTransaction method 73

S

setAccount method 353
setAccountBalanceDate method 154
setAccountID method 205, 260, 261, 274
setAccountType method 154, 353
setAcctID method 219, 244
setAction method 219
setAddress method 331, 354
setAddress1 method 310
setAddress2 method 310
setAddress3 method 311
setAlias method 365, 380
setApplicationName method 414
setApplName method 219
setBalanceDate method 292
setBillerID method 155, 396, 407
setBillerStatus method 331
setBillingAddress method 155

setBillingContext method 166, 392
setBillType method 206
setBrand method 381
setBusinessCategory method 331
setCaseNum method 245
setCharSet method 332, 354
setCity method 311
setContext method 73
setCreatedBy method 220, 332, 354
setCsrName method 245
setCurrency method 332, 355
setCurrentBalance method 293
setCurrentCharge method 293
setCustomerStatus method 355
setCustomerType method 155
setDayPhone method 355
setDescription method 155, 189, 209, 293, 332, 356, 366, 381, 382
setDiscountBeforeDate method 294
setDiscountRate method 294
setDiscountType method 294
setDisputeAmount method 294
setDisputeStatus method 156
setDocID method 245, 260, 261, 294
setDocStatus method 206
setDocType method 207
setEmail method 333, 356
setEventData method 407
setEventDesc method 388
setEventHandler method 397
setEventHandlerStatus method 397
setEventID method 388
setEventName method 388, 397, 398, 408
setEventPopDate method 408
setEventPriority method 408
setEventProcessDate method 408
setEventPushDate method 409
setEventStatus method 409
setExtension method 156, 207, 246, 261, 295, 333, 356
setFaxNumber method 333, 356, 357
setFileID method 207, 209, 261, 295, 296
setFirstName method 357
setFirstViewDate method 208

setFrequency method 275
setGroup method 357
setHeader method 190, 191, 212, 213, 228, 229, 230, 235, 236, 248, 249, 264, 265, 278, 279, 299, 300
setLanguage method 333, 357
setLastName method 358
setLastViewDate method 208
setLateAfterDate method 296
setLoginName method 358
setMaxAmount method 381
setMerchantID method 334
setMessage method 220, 246
setMiddleName method 358
setMinAmount method 381
setMsgFrom method 246
setMsgStatus method 246
setNickName method 358
setNightPhone method 359
setOrgID method 220
setOrgName method 334, 359
setOriginalAmount method 297
setPassword method 359
setPasswordHint method 359
setPasswordLastChanged method 360
setPayAmount method 275, 276
setPayAmountType method 276
setPaymentHandle method 366
setPaymentType method 261
setPaytermsCode method 297
setPenaltyType method 298
setPhone method 334
setPmtAcctID method 276
setPmtAmount method 262
setPmtMode method 262
setPmtTxnID method 262
setPmtTxnStatus method 263
setPostalCode method 311
setPreviousBalance method 298
setPrimaryContact method 334, 360
setProcessor method 382
setProfile method 156
setProfileExtension method 335, 360
setProperty method 65

setSecondaryContact method 335, 361
setSecretWord method 361
setSenderType method 247
setServiceAddressID method 156
setServiceExtension method 335, 361
setServiceOptions method 157
setSince method 157
setStartOnDate method 277
setState method 312
setStatus method 74, 371
setStopOnDate method 277
setSubject method 247
setSuffix method 361
setSumStatus method 299
setTaxID method 157, 335
setTimeZone method 336, 362
setType method 247, 312, 336, 366, 382
setUnpaidBalance method 157
setUnpaidBills method 158
setURL method 299, 336
setViewCount method 208

T

toLDAPOrderBy method 111
toOrderBy method 111
top method 120, 121

U

updatePassword method 362

W

website
for iPlanet customer support 22

X

XML tag constants 56

