



SOAP API 11.3

for LexisNexis® Bridger Insight® XG (XG) 5 Web Services



The recipient of this material (hereinafter “the Material”) acknowledges that it contains confidential and proprietary data the disclosure to, or use of which by, third parties will be damaging to LexisNexis Risk Solutions and its affiliated companies (hereinafter “LexisNexis”). Therefore, recipient agrees to hold the Material in strictest confidence, not to make use of it other than for the purpose for which it is being provided, to release it only to employees requiring such information, and not to release or disclose it to any other party. Upon request, recipient will return the Material together with all copies and modifications, if any, to LexisNexis.

LexisNexis shall not be liable for technical or editorial errors or omissions contained herein. The information in this publication is subject to change without notice and is provided “as-is” without a warranty of any kind. Nothing herein should be construed as constituting a warranty, as any applicable warranty is exclusively contained in your signed agreement with LexisNexis.

All names, addresses, birth dates or other identifying information in the text, or on the sample reports and screens shown in this document, are of fictitious persons and entities and have been created for illustrative purposes only. Any similarity to the name of any real person, school, business, other entity, address, date of birth, or other identifying information is purely coincidental.

LexisNexis and the Knowledge Burst logo are registered trademarks of RELX Inc. Bridger Insight is a registered trademark of LexisNexis Risk Solutions Inc. FraudPoint and InstantID are registered trademarks of LexisNexis Risk Solutions FL Inc. WorldCompliance is a trademark of World Compliance Inc. Other products and services may be trademarks or registered trademarks of their respective companies.

© 2020 LexisNexis Risk Solutions Group

Contents

Chapter 1: SOAP API.	9
API Version.	10
Searchable Data.	10
About This Publication.	11
Audience.	11
Content.	11
Conventions.	12
Publication Suggestions.	12
Support and Resources.	12
LexisNexis Technical Support.	12
LexisNexis Technical Support Website.	13
Training.	13
Chapter 2: Web Service Setup.	14
Communication Protocols and Programming Languages.	14
Communication Architecture.	14
Proxy Objects.	15
Security.	15
Rate Limits.	17
Bridger Insight XG Access.	17
Web Service Planning.	18
Chapter 3: Account Interface.	19
ChangePassword Method.	19
GetDaysUntilPasswordExpires Method.	20
ClientContext Class.	21
DPPA Permissible Uses.	22
GLBA Permissible Uses.	24
Chapter 4: Search Interface.	26
Data File Management.	26
GetDataFileList Method.	26

DataFileInfo Class.	27
Search Method.	28
Search Request Classes.	29
Search Request Class Hierarchy.	29
Search Configuration Classes.	30
Search Input Classes.	51
Search Response Classes.	56
Search Response Class Hierarchy.	56
Record Information Classes.	60
FraudPoint Score Result Classes.	67
InstantID Result Classes.	70
List Screening Result Classes.	79
Chapter 5: Results Interface.	92
Run Management.	92
DeleteRuns Method.	92
GetRunInfo Method.	93
SearchRuns Method.	94
Run Management Classes.	94
Result Management.	99
DeleteResultRecords Method.	99
GetResultRecords Method.	100
SearchResultRecords Method.	101
SetRecordState Method.	101
Result Management Classes.	102
Attachment Management.	118
AddAttachment Method.	119
DeleteAttachment Method.	119
GetAttachment Method.	120
Attachment Management Class.	121
Chapter 6: List Maintenance Interface.	122
List Management.	122

AddList Method.....	123
DeleteList Method.....	123
GetList Method.....	124
IndexList Method.....	125
SearchLists Method.....	126
UpdateList Method.....	126
List Management Classes.....	127
List Record Management.....	130
AddRecords Method.....	130
AddResultRecord Method.....	131
DeleteRecords Method.....	132
GetRecords Method.....	133
SearchListRecords Method.....	134
UpdateRecord Method.....	135
List Record Management Classes.....	136
Chapter 7: EFT Parsed Input Classes.....	148
ACH Parsed Input Classes.....	148
ParsedACH Class.....	148
ParsedACHAdditionalInfo Class.....	149
ParsedACHAddress Class.....	149
ParsedACHField Class.....	150
ParsedACHID Class.....	150
ParsedACHNote Class.....	150
ParsedACHParty Class.....	151
EFT Party Types.....	152
Fedwire Parsed Input Classes.....	153
ParsedFedwire Class.....	154
ParsedFedwireAdditionalInfo Class.....	154
ParsedFedwireAddress Class.....	154
ParsedFedwireAmount Class.....	155
ParsedFedwireField Class.....	156

ParsedFedwireID Class.	156
ParsedFedwireNote Class.	156
ParsedFedwireParty Class.	157
ParsedFedwirePhone Class.	158
ISO 20022 Parsed Input Classes.	158
ParsedISO20022 Class.	158
ParsedISO20022AdditionalInfo Class.	159
ParsedISO20022Address Class.	159
ParsedISO20022Field Class.	160
ParsedISO20022ID Class.	160
ParsedISO20022Note Class.	161
ParsedISO20022Party Class.	161
ParsedISO20022Phone Class.	162
SWIFT Parsed Input Classes.	163
ParsedSWIFT Class.	163
ParsedSWIFTAdditionalInfo Class.	164
ParsedSWIFTAddress Class.	164
ParsedSWIFTAmount Class.	165
ParsedSWIFTField Class.	165
ParsedSWIFTID Class.	166
ParsedSWIFTNote Class.	166
ParsedSWIFTOriginalMessage Class.	166
ParsedSWIFTParty Class.	167
Chapter 8: Entity Information.	168
Input Entity Classes.	168
Date Class.	168
InputAdditionalInfo Class.	169
InputAddress Class.	169
InputID Class.	170
InputName Class.	172
InputPhone Class.	172

Entity Record Classes.	173
Date Class.	173
EntityAdditionalInfo Class.	174
EntityAddress Class.	174
EntityAKA Class.	175
EntityID Class.	176
EntityName Class.	176
EntityPhone Class.	177
Entity Criteria Classes.	177
EntityCriteriaAdditionalInfo Class.	178
EntityCriteriaAddress Class.	178
EntityCriteriaID Class.	179
Possible Type Values.	179
Possible Additional Information Types.	180
Possible ID Types.	180
Chapter 9: Errors.	181
ServiceFault Class.	181
ServiceFault Error Messages.	181
ResultError Class.	187
Error Messages.	188
Error Codes.	189
Appendix A: C# Code Examples.	192
AddRecord Code Example.	192
AddResultRecord Code Example.	193
DeleteRecords Code Example.	194
UpdateRecord Code Example.	195
GetRecords Code Example.	196
SearchListRecords Code Example.	197
UpdateList Code Example.	198
SearchLists Code Example.	199
AddList Code Example.	200

IndexList Code Example.	201
DeleteList Code Example.	201
GetList Code Example.	202
ChangePassword Code Example.	203
GetDaysUntilPasswordExpires Code Example.	204
Search Code Example.	205
GetDataFileList Code Example.	206
GetRunInfo Code Example.	207
SearchRuns Code Example.	208
DeleteRuns Code Example.	209
DeleteResultRecords Code Example.	209
GetResultRecords Code Example.	210
SetRecordState Code Example.	211
SearchResultRecords Code Example.	212
AddAttachment Code Example.	213
DeleteAttachment Code Example.	214
GetAttachment Code Sample.	215
Index.	217

Chapter 1

SOAP API

The SOAP (Simple Object Access Protocol) API (application programming interface) for LexisNexis[®] Bridger Insight[®] XG (XG) web services lets you incorporate much of the XG functionality into your existing systems. This functionality includes searching, result management, and list management.

The SOAP API is a single WCF (Windows Communication Foundation) service with the following WSDL (Web Services Description Language) interfaces:

- The Account interface lets you manage XG passwords.
- The Search interface lets you search certain search services.
- The Results interface lets you query and manage alerts and batch runs.
- The List Maintenance interface lets you query and manage accept lists and custom screening lists.

The SOAP API accepts XML messages over SOAP. The WSDL defines the SOAP interface and allows tools to create proxy objects to generate and invoke the appropriate SOAP methods. These SOAP methods are then sent over TCP or HTTPS. For each request, a method defines an action that is performed either by or upon an instance of a class. The response also returns information in classes.

The WSDL has been tested with Microsoft Visual Studio and SoapUI. If using Python, spaces in web addresses need to be escaped. Other programs may also be able to consume the WSDL.

The WSDL is often used in combination with SOAP and XML Schema to provide web services over the internet. A client program that connects to a web service can read the WSDL to determine what functions are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL. If your staff has limited SOAP knowledge, you can use a third-party tool to create client proxy objects from the WSDL.

The way that you access the SOAP API depends on your product and environment:

XG Service

Depending on your account, you access the WSDL at: `https://<Your XG Environment>/LN.WebServices/<web service version>/XGServices.svc?wsdl`.

XG Enterprise

The WSDL location is configured when you install the product. See the *LexisNexis[®] Bridger Insight[®] XG Enterprise Installation Manual*.

The environment is displayed in the browser address bar for the XG user interface.

API Version

Each version of the SOAP API provides the functionality for a certain XG product version.

API Alignment with Bridger Insight XG Functionality

SOAP API Version	Product Functionality
8.0	XG 3.4
9.0	XG 4.1-4.4
10.0	XG 4.5
10.1	XG 4.6
10.2	XG 4.6 and the False Match Update functionality for name and DOB (date of birth)
10.3	Additional functionality includes the InputAccount class
11.0	XG 5.0
11.1	XG 5.0 and the false match update functionality for name, DOB, address, AKA (also known as), citizenship, ID number, gender, and phone number
11.2	Additional functionality includes the ability to provide a provider ID or account type as entity information and to exclude list match information from the result
11.3	Additional functionality includes duplicate match suppression and LexisNexis® Intelligent Match Decision Solution (IMDS)

You can use earlier versions of the SOAP API. However, new functionality is not available in these versions.

Searchable Data

XG lets you search either structured input data or unstructured input data.

Structured data is organized into defined fields that lets XG treat like information the same for more relevant results. You can use custom formats to search business entity records, individual entity records, or unknown entity records. You can also use the following corresponding default formats to search payment transaction data:

ACH (Automated Clearing House) files

For information about ACH record content and format, see <https://www.nacha.org>.

For information about supported ACH formats and the data processed by XG, see the knowledge base article on the LexisNexis® Technical Support website at https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1515.

Fedwire files

For information about Fedwire record content and format, see <https://www.frbervices.org>.

For information about supported Fedwire formats and the data processed by XG, see the knowledge base article on the LexisNexis Technical Support website at https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1516.

ISO (International Organization for Standardization) 20022 files

For information about ISO 20022 record content and format, see <https://www.iso20022.org>.

For information about supported ISO 20022 formats and the data processed by XG, see the knowledge base article on the LexisNexis Technical Support website at https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1549.

SWIFT (Society for Worldwide Interbank Financial Telecommunication) files

For information about SWIFT record content and format, see <https://www.swift.com>.

For information about supported SWIFT formats and the data processed by XG, see the knowledge base article on the LexisNexis Technical Support website at https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1514.

Unstructured data is not organized into defined fields. You can search unstructured data as a text record or by using the Unstructured format. A text record contains one required field of unstructured text and optional ID fields. When you search unstructured data, XG must search files one text string at a time. This type of search may yield different results than when XG searches a file with identified fields.

About This Publication

This publication provides information about how to use XG web services. References to other sections may be active hyperlinks. You can click the link to take you to the referenced section.

Audience

This publication contains technical information and is geared primarily for C# programmers. You should also have a good understanding of XML, SOAP, classes, methods, and WSDL. Web services should be used by staff with the necessary technical expertise. If not set up appropriately, your system could become unusable.

Content

This publication is primarily organized by its web service interfaces: Account, Search, Results, and List Maintenance.

Conventions

This publication uses various conventions to provide specific information. These conventions are described in the following sections.

Information Icons

LexisNexis[®] Risk Solutions uses icons to highlight specific types of information. uses icons to highlight specific types of information.



This note icon precedes important information.



This tip icon precedes helpful information.



This warning icon precedes information that is intended to help prevent you from negatively affecting your processes.

Publication Suggestions

Your suggestions for publication improvements are welcome.

Send comments or suggestions to: TechnicalPublications@lexisnexisrisk.com.

Support and Resources

Support and resources are available to XG clients who have purchased the product. Support and resources are not available to trial users.

LexisNexis Technical Support

Support is included with all XG contracts. The level of support that is provided varies based on the complexity of the XG installation and the contracted terms.

For more information about the contracted terms, contact your LexisNexis sales representative.

To access the knowledge base, Live Chat, Ask a Question, support hours, and other support services, you must first register for a free user account on the LexisNexis Technical Support website.



To create an account, you need your XG product code that LexisNexis provided to your organization.

LexisNexis Technical Support Website

The LexisNexis Technical Support website provides access to knowledge base articles, a documentation library, training courses, and web-based support.

You access the LexisNexis Technical Support website at <https://bridgerinsight.custhelp.com>. You use the website to complete the following tasks:

- Search the knowledge base and support history.
- Submit, track, and update your product questions.
- Set up a profile of contact information and notifications.
- Download guides, white papers, and other documentation.
- View training courses.

Training

XG training is available at the LexisNexis Technical Support website at no additional cost.

The website lets your customer contact view self-paced courses or register for instructor-led courses that are delivered by webinar.

Based on contracted terms, your customer contact can also schedule a remote or on-site training session with your TAM (technical account manager) to refresh your end users, customer contacts, and administrators about XG features and processes. For more information, contact LexisNexis Technical Support.

Chapter 2

Web Service Setup

LexisNexis[®] Bridger Insight[®] XG (XG) web services provide setup flexibility. Before you implement the services, consider your environment and business requirements.

Communication Protocols and Programming Languages

XG web services supports HTTP/HTTPS as the communication protocols. The HTTP/HTTPS web service is hosted in IIS (Internet Information Services) and exposes both a WCF (Windows Communication Foundation) MEX (metadata exchange) endpoint and an HTTP GET method.

You can use any programming language that can consume a WCF MEX endpoint or a WSDL (Web Services Description Language).

Communication Architecture

You can add additional application servers and IIS servers to help ensure that if one server fails, there is still one or more that can handle client requests.

However, if there are multiple servers, then a load balancer is required to distribute client requests between servers. IIS communicates with one or more application servers through the TCP/IP interface.

Depending upon load, adding additional application servers and IIS servers into the system may increase performance. However, if there are multiple servers, then a load balancer is required to distribute client requests between servers. Load balancing may be done through hardware, software, or internal load balancing.

Proxy Objects

If you use a programming language that can consume a WCFMEX endpoint, client proxies can be auto-generated or can be created through `ChannelFactory<IAccount>`, `ChannelFactory<IListMaintenance>`, `ChannelFactory<IResults>`, or `ChannelFactory<ISearch>`.

Examples of auto-generated proxies are those generated by `svcutil.exe` (or a similar tool) or those generated when a Service Reference or Web Reference is added to a project in Visual Studio.

For best performance when auto-generated proxies are used, reuse the proxy object. Be sure to develop code to handle instances when the proxy object channel is in a faulted state.

It may not always be possible to reuse the proxy object. For example, while the proxy object is thread safe there can be contention issues if you use it in multiple threads. It is possible to get good performance if you take advantage of the proxy object constructors that don't take `Binding` as an argument and if you do not use the `ChannelFactory`, `Endpoint`, and `ClientCredentials` properties. This allows for the underlying code to internally cache the `ChannelFactory` object, which makes for efficient creation of proxy objects. If this is not possible, then consider using proxy pooling.

For best performance when `ChannelFactory<IAccount>`, `ChannelFactory<IListMaintenance>`, `ChannelFactory<IResults>`, or `ChannelFactory<ISearch>` is used, create a single instance of the channel factory and then use its `CreateChannel` method to create client proxy objects.

If you use a programming language that can consume a WCFMEX endpoint, client proxy objects (either auto-generated or created through `ChannelFactory<IAccount>`, `ChannelFactory<IListMaintenance>`, `ChannelFactory<IResults>`, or `ChannelFactory<ISearch>`) are thread safe. However, there can be some contention issues when the same object is used in multiple threads.

Security

XG Enterprise clients can configure the security tag for XG web services bindings. This setting is not available to XG Service clients.

This configuration is typically completed when XG Enterprise is installed. The equivalent settings must also be set in the application and IIS.

For the `mode` attribute in the security tag, you may specify one of the following values:

None

Messages are not secured in any way. It is the fastest option, but only use it if messages do not need to be securely transmitted between the client and the server.

Message

Messages are secured at the message level using WS-Security and other specification. It is transport-protocol independent, more extensible, and ensures end-to-end security. However, it is several times slower than transport security.

Transport

Messages are secured at the transport level using, for example, HTTPS. This option is widely adopted and less complex. However, it provides only point-to-point security.

TransportWithMessageCredential

Messages are secured using message security to authenticate the client and transport security to authenticate the server and provide message security. This option is only slightly slower than the transport mode. However, like transport mode, it provides only point-to-point security.

End-to-end security means that the message is secure even when routed through intermediary application nodes. Point-to-point security means that the message is secure only from one application node to the next application node so intermediary application nodes have access to the message.

With the `basicHttpBinding` binding used in the `web.config` file if you use the HTTP protocol, you must use the mode `None`. If you use the HTTPS protocol, you must use the mode `Transport`. The installer sets these appropriately. The `web.config` file is located in the XG installation folder.

With the `netTcpBinding` binding used in the `BridgerInsightService.exe.config` file, by default the mode is set to `None`. The `BridgerInsightService.exe.config` file is located in the XG installation folder. If the security mode is `Transport`, the security tag must contain a transport tag (Message by default) and the `clientCredentialType` value needs to be one of the following:

None

The WCF service does not authenticate the client.

Windows

Clients are authenticated against Active Directory using a Windows token. The service is authenticated by the Windows credentials or by the SSL (secure sockets layer) certificate if using HTTPS.



There are other possible client credential types, but they have not been tested with XG web services.

Rate Limits

XG web services supports rate limitations for improved system stability and scalability. Rate limits also help protect against slow performance and denial-of-service attacks for systems that are accessible outside of an organization's firewall.

XG Enterprise clients can use the following value to enable this feature in the installation configuration files: `<globalSymbol name="EnableApiRateLimit">false</globalSymbol>`. The feature must also be configured for each client instance in Enterprise Manager.

The following response headers are returned when rate limiting is enabled.

X-RateLimit-Limit

The header value is an integer that shows how many requests are allowed per period.

X-RateLimit-Remaining

The header value is an integer that shows how many requests can be submitted in the current period before rate limiting is enforced and requests will start to fail.

X-RateLimit-Reset

The header value is an integer that shows how many seconds until the period expires and a new timer begins. If you have reached your rate limit, you cannot submit requests until the period expires and a new timer begins.

When the rate limit is surpassed, the web service response returns the following error: The number of requests received has exceeded the configured rate limit of # per (second|minute|hour).

Bridger Insight XG Access

XG web services requires users to provide certain credentials for access.

You must provide the following credentials:

Client ID

The client ID is required for most product setups. The client ID identifies the XG client instance of the user who sent the request.

User ID

The user ID is required if your system does not use Windows authentication.

Password

The XG user password is required if your system does not use Windows authentication.

In the Account interface, you can use the `GetDaysUntilPasswordExpires` method to determine when the password expires. You can use the `ChangePassword` method to change the password.

XG passwords, along with their associated user accounts, can also be managed in the XG interface. For more information, see the *LexisNexis® Bridger Insight® XG Help*.

DPPA (Driver's Privacy Protection Act) permissible use

A DPPA permissible use is required for LexisNexis® FraudPoint® Score searches and LexisNexis® InstantID® searches on individual entities.

GLBA (Gramm-Leach-Bliley Act) permissible use

A GLBA permissible use is required for FraudPoint Score searches and InstantID searches.

The user that sends the request must have the division access and the role access that is required to perform the actions. For more information, see the *LexisNexis® Bridger Insight® XG Help*.

Web Service Planning

You should consider a number of factors to best implement XG web services.

At a minimum, you must determine the following information:

- How to implement the service based on your existing system architecture
- What language to code in
- What protocols to use
- How to use proxy objects
- How to implement the service based on your security requirements
- What input information you want to search
- What type of searches you want to perform
- Which functions you should include in the call
- How you want to manage results and logging
- What load and response time requirements are anticipated
- How to manage system maintenance and password changes
- How to manage error handling and problem alerting
- Which user account calls the methods

Chapter 3

Account Interface

LexisNexis[®] Bridger Insight[®] XG (XG) web services provides an Account interface that you can use to manage XG passwords.

User accounts, passwords, the password length, and the password expiration period are managed in the XG interface. For more information, see the *LexisNexis[®] Bridger Insight[®] XG Help*.

The Account interface lets you complete the following tasks:

- Change the password that is associated with an XG user account.
- Get the number of days until a password expires.

Related Information

[C# Code Examples](#) on page 192

ChangePassword Method

The ChangePassword method changes the XG password that is associated with the user ID.

Syntax

```
bool ChangePassword(ClientContext context, string newPassword)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

newPassword

The parameter contains the new XG password.

Passwords are case sensitive.



Passwords should be secure and confidential.

The password must meet the following requirements:

- The password must contain at least one letter, number, and special character. Examples of special characters include !, @, or #.
- The password cannot contain the user ID or its reverse.
- The password cannot be one of the previous 12 passwords.

Return Values

The response always returns True.

If the method fails or there is an error, then the response also returns a ServiceFault class.

Related Information

[ClientContext Class](#) on page 21

[ServiceFault Class](#) on page 181

[ChangePassword Code Example](#) on page 203

GetDaysUntilPasswordExpires Method

The GetDaysUntilPasswordExpires method returns the number of days until the XG password expires.

Syntax

```
int GetDaysUntilPasswordExpires(ClientContext context)
```

Parameters

The method contains the following parameter:

context

The ClientContext class contains the XG client information and the XG session information.

Return Values

The response returns the number of days until the password expires.

Related Information

[ClientContext Class](#) on page 21

[GetDaysUntilPasswordExpires Code Example](#) on page 204

ClientContext Class

The ClientContext class contains the XG client information and the XG session information.

This class is required for all requests sent from any XG web services interface.



For XG 5, your organization can create a user account that is only used for web services. Your system must have IP settings enabled. The password for the user account never expires.

Class Properties

Name	Type	Description
ClientID	String	This property contains the client ID that identifies the XG web services instance that sent the request. This property is required.
ClientReference	String	This property contains optional user-defined information to identify the request. In the Search interface, this property is returned in the SearchResult class.
DPPA	DPPAChoiceType	This property contains the DPPA (Driver's Privacy Protection Act) permissible use. Possible values: • NA • Choice0 • Choice1 • Choice2 • Choice3 • Choice4 • Choice6 The default value is NA. This value is not valid if a DPPA permissible use is required. This property is required for LexisNexis® FraudPoint® Score searches or LexisNexis® InstantID® searches on individual entities.

Class Properties (Continued)

Name	Type	Description
GLB	Integer	This property contains the GLBA (Gramm-Leach-Bliley Act) permissible use. Possible values: • NA • 0 • 1 • 3 • 5 • 6 • 7 • 12 The default value is NA. This value is not valid if a GLBA permissible use is required. This property is required for FraudPoint Score searches or InstantID searches.
Password	String	This property contains the password of the user who sends the request. Passwords are case sensitive. This property is required if Windows authentication is not set up.
UserID	String	This property contains the user ID that identifies the user who sends the request.  <i>The user must have the division access and the role access that is required to perform the requested actions. See the LexisNexis® Bridger Insight® XG Help.</i> This property is required if Windows authentication is not set up.

Related Information

[SearchResults Class](#) on page 67

DPPA Permissible Uses

Users must set a DPPA permissible use for each individual search against LexisNexis® FraudPoint® or InstantID.



By entering your response and proceeding with your search, you certify to LexisNexis® Risk Solutions that you are in, and assume full responsibility for, compliance with DPPA in accordance with your response herein and you agree to indemnify, defend, and hold harmless LexisNexis Risk Solutions from any breach of such certification by you, your agents, or contacts and any damages,

fees, and costs associated therewith. Information from this file may not be used to determine a customer's eligibility for credit, insurance, or employment pursuant to the Fair Credit Reporting Act, (15 USC Sec. 1681).

Users can set one of the following possible DPPA permissible uses. The service searches against driver licenses, motor vehicle registrations, and boat registrations for the states that recognize the permissible use that you set. If you set **No Permissible Use**, the service does not search driver licenses, motor vehicle registrations, or boat registrations.

DPPA Permissible Uses

Web Service Value	Name	Description
Choice0	No Permissible Use	For persons who do not have a DPPA permissible use.
Choice1	Government Agency or On Government Behalf	For use by any government agency, including any court or law enforcement agency, in carrying out its functions, or any private person or entity acting on behalf of a Federal, State, or local agency in carrying out its functions.
Choice2	Motor Vehicle or Driver Safety	For use in connection with matters of motor vehicle or driver safety and theft; motor vehicle emissions; motor vehicle product alterations, recalls, or advisories; performance monitoring of motor vehicles, motor vehicle parts and dealers; motor vehicle market research activities, including survey research; and removal of non-owner records from the original owner records of motor vehicle manufacturers.
Choice3	Debt Recovery/Fraud	For use in the normal course of business by a legitimate business or its agents, employees, or contractors, but only—(A) to verify the accuracy of personal information submitted by the individual to the business or its agents, employees, or contractors; and (B) if such information as so submitted is not correct or is no longer correct, to obtain the correct information, but only for the purposes of preventing fraud by, pursuing legal remedies against, or recovering on a debt or security interest against, the individual.

DPPA Permissible Uses (Continued)

Web Service Value	Name	Description
Choice4	Litigation	For use in connection with any civil, criminal, administrative, or arbitral proceeding in any Federal, State, or local court or agency or before any self-regulatory body, including the service of process, investigation in anticipation of litigation, and the execution or enforcement of judgments and orders, or pursuant to an order of a Federal, State, or local court.
Choice6	Insurer	For use by any insurer or insurance support organization, or by a self-insured entity, or its agents, employees, or contractors, in connection with claims investigation activities, antifraud activities, rating or underwriting.

GLBA Permissible Uses

Users must set a GLBA exception (permissible use) for each search against FraudPoint Score or InstantID.



Misrepresenting your access is a violation of the LexisNexis Risk Solutions subscriber agreement and may be a violation of federal and state laws. Any use of LexisNexis Risk Solutions information other than for the selected purpose is grounds for account termination and may be referred to the FTC (Federal Trade Commission), law enforcement, or the appropriate state investigative agency.

Users can set one of the following possible GLBA permissible uses. The service searches the data that is regulated by the GLBA that you set and the data that is not regulated by the GLBA. If you set **No Permissible Use**, the service searches the data that is not regulated by the GLBA. The use of information obtained from these databases is limited to use with the ordinary course and scope of the user's business or profession. The user represents and warrants that such use is for lawful and appropriate purpose.

GLBA Permissible Uses

Web Services Value	Name	Description
0	No Permissible Use	For persons who do not have a GLBA permissible use.
0	Resolving Customer Disputes or Inquiries	For use in resolving customer disputes or inquiries.

GLBA Permissible Uses (Continued)

Web Services Value	Name	Description
0	As allowed by the Right to Financial Privacy Act of 1978	To the extent specifically permitted or required under other provisions of laws and in accordance with the Right to Financial Privacy Act of 1978, to law enforcement agencies, self-regulatory organizations or for an investigation on a matter related to public safety.
1	Transactions Authorized by Consumer	As necessary to effect, administer, or enforce a transaction requested or authorized by the consumer.
3	Persons with a Legal/Beneficial Interest in the Consumer	Use by persons, or their representatives, holding a legal or beneficial interest relating to the consumer.
5	Fraud Prevention and Detection	To protect against or prevent actual or potential fraud, unauthorized transactions, claims or other liability.
6	Required Institutional Risk Control	In required institutional risk control programs.
7	Legal Compliance	In complying with federal, state, or local laws, rules, and other applicable legal requirements.
12	Persons Acting in a Fiduciary Capacity for the Consumer	Use by persons acting in a fiduciary or representative capacity on behalf of the consumer.

Chapter 4

Search Interface

LexisNexis[®] Bridger Insight[®] XG (XG) web services provide a Search interface that you can use to perform searches on your input information.

You can choose to write search results to your XG database for further processing. You can access these alerts through the Results interface or through the XG application.

The Search interface lets you view information about available accept lists and screening lists.

The Search interface also lets you search the following search services:

- LexisNexis[®] FraudPoint[®] Score
- LexisNexis[®] InstantID[®]
- LexisNexis[®] List Screening

You can configure the settings that XG uses to search your input information by specifying a predefined search that was created in the XG application. Or, you can provide search configuration settings in the request.

Data File Management

For search availability and management purposes, the Search interface uses the GetDataFileList method to view information on accept lists, custom screening lists, and LexisNexis screening lists.

Related Information

[C# Code Examples](#) on page 192

GetDataFileList Method

The GetDataFileList method returns a list of the available accept lists, custom screening lists, and screening lists.

Syntax

```
List<DataFileInfo> GetDataFileList(ClientContext context)
```

Parameters

The method contains the following parameter:

context

The ClientContext class contains the XG client information and the XG session information.

Return Values

The DataFileInfo class contains the information about the accept list, the custom screening list, or the LexisNexis screening list.

Related Information

[DataFileInfo Class](#) on page 27

[ClientContext Class](#) on page 21

[GetDataFileList Code Example](#) on page 206

DataFileInfo Class

The DataFileInfo class contains the information about the accept list, the custom screening list, or the LexisNexis screening list.

Class Properties

Name	Type	Description
Custom	Boolean	True indicates that the file is a custom screening list.
Description	String	This property contains the list description.
FileDate	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that LexisNexis provided the file.
HasWeakAKAs	Boolean	True indicates that the file has low-quality AKA (also known as) information.
ID	GUID	This property contains the system-generated number that identifies the list.
Name	String	This property contains the list name.
PublishedDate	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that the source agency provided the file.

Class Properties (Continued)

Name	Type	Description
SearchCriteria	String	This property contains the delimited list of the search criteria that is available for the file. You cannot include this entire list in a search request. For more information about search criteria, see https://bridgerinsight.custhelp.com/app/answers/detail/a_id/882 .
Type	DataFileType	This property contains the data file type. Possible values: • UDF (XG file) • ADF (accept list) • BDF (entity list) • CDF (country list) • SDF (XG file)

Search Method

The Search method sends the input information to the XG server for processing using the specified search settings.

Syntax

```
SearchResults Search(ClientContext context, SearchConfiguration config, SearchInput input)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

config

The SearchConfiguration class contains the settings to be used in a search.

input

The SearchInput class contains the input records.

Return Values

The SearchResults class contains the search results.

Related Information

[SearchConfiguration Class](#) on page 33

[SearchInput Class](#) on page 55

[SearchResults Class](#) on page 67

[Search Code Example](#) on page 205

Search Request Classes

In the search request, you can include the configuration settings that you want XG to use in the search and the input information that you want to search.

Some classes contain other classes as properties. The classes are described in alphabetical order. Classes that are used in multiple web services interfaces may be described in another section.

Related Information

[Search Configuration Classes](#) on page 30

[Search Input Classes](#) on page 51

Search Request Class Hierarchy

A class hierarchy depicts the structure of the classes within the code.

The following class hierarchy depicts the structure of the search request:

"ClientContext Class" on page 21

"SearchConfiguration Class" on page 33:

- "AssignmentInfo Class" on page 31
- "FraudPointConfiguration Class" on page 31
- "InstantIDConfiguration Class" on page 33
- "WatchlistConfiguration Class" on page 41:
 - "WatchlistAutomaticFalsePositiveRules Class" on page 35
 - "WatchlistDataFile Class" on page 40 > "WatchlistSearchCriteria Class" on page 50 > "WatchlistSearchCriteriaGroup Class" on page 51
 - "WatchlistFalseMatchSettings Class" on page 43
 - "WatchlistGeneralOptions Class" on page 44
 - "WatchlistIntelligentMatchDecisionSolution Class" on page 46
 - "WatchlistMatchDispositionRules Class" on page 47
 - "WatchlistMatchOptions Class" on page 49 > "WatchlistEFTMatchOptions Class" on page 42

"SearchInput Class" on page 55 > "InputRecord Class" on page 54:

- "InputBasicID Class" on page 52
- "InputEFT Class" on page 52
- "InputEntity Class" on page 53:
 - "InputAdditionalInfo Class" on page 169 > "Date Class" on page 95
 - "InputAddress Class" on page 169
 - "InputID Class" on page 170 > "Date Class" on page 95
 - "InputName Class" on page 172
 - "InputPhone Class" on page 172
- "InputText Class" on page 55 > "InputID Class" on page 170 > "Date Class" on page 95

Search Configuration Classes

The search request includes the classes that contain the search configuration settings.

Related Information

[Search Request Class Hierarchy](#) on page 29

AssignmentInfo Class

The AssignmentInfo class sets division access and assigns alerts to specified roles or users. You can also restrict record access to a division.



Divisions, roles, and users are set up through the XG application. See the LexisNexis® Bridger Insight® XG Help.

Class Properties

Name	Type	Description
Division	String	This property restricts alert access to the specified division. Only users in the specified division can process the alerts generated from the search. This property is if your organization has created divisions.
EmailNotification	Boolean	True sends a single email message to each specified user or users in specified roles who were assigned alerts for each completed request.  <i>If the results are restricted to a division, only users within a role that has access to that division receive an email message.</i> This property is required.
RolesOrUsers	List<String>	This property contains the roles or the users to be assigned alerts. You must specify either roles or users, not both. If you specify users, you must specify the user ID and not the user name. If you specify multiple roles, the alerts are assigned to all specified roles. Only the users within the roles who belong to the specified division can process the alerts. If you specify multiple users, the alerts are assigned in a round-robin fashion to specified users. If you included a Division property, only the users who belong to the specified division are assigned alerts.
Type	AssignmentType	This property contains the assignment type. Possible values: • Role • Unknown • User The Unknown value is not valid in a request.

FraudPointConfiguration Class

The FraudPointConfiguration class contains the FraudPoint Score configuration settings.

Class Properties

Name	Type	Description
FriendlyFraudIndex	Integer	This property sets XG to generate an open alert if the index is greater than or equal to the specified value. The index indicates the identity profile is at risk for being a victim of identity fraud perpetrated by a close associate. For example, the individual has high-risk relatives and close associates. The index range is 1-9.
ManipulatedIdentityIndex	Integer	This property sets XG to generate an open alert if the index is greater than or equal to the specified value. The index indicates the risk that the identity is being intentionally manipulated. For example, there is a suspicious amount of variation of the PII (personally identifiable information) associated with this identity. The index range is 1-9.
Password	String	The FraudPoint Score password. This property is required.
UserID	String	The FraudPoint Score user ID. This property is required.
StolenIdentityIndex	Integer	This property sets XG to generate an open alert if the index is greater than or equal to the specified value. The index indicates the risk that the transaction is in connection with a real identity that has been compromised. For example, the identity is applying with a new address that is not likely associated with the true owner of the identity. The index range is 1-9.
SuspiciousActivityIndex	Integer	This property sets XG to generate an open alert if the index is greater than or equal to the specified value. The index indicates the level of suspicious and high-risk activity that was previously observed for the input elements. For example, the input elements are tied to records that were previously deemed to be suspicious or high risk. The index range is 1-9.
SyntheticIdentityIndex	Integer	This property sets XG to generate an open alert if the index is greater than or equal to the specified value. The index indicates the risk that the input identity is a synthetic identity. For example, this identity is only reported by credit bureaus or is missing characteristics of normal identities. The index range is 1-9.

Class Properties (Continued)

Name	Type	Description
Version	FraudPointVersion	The LexisNexis® FraudPoint® version to be used to perform the search. Possible values: • None • V1_0 • V2_0
VulnerableVictimIndex	Integer	This property sets XG to generate an open alert if the index is greater than or equal to the specified value. The index indicates the identity profile is at risk for being a victim of identity fraud. For example, a high-occupancy location or minors. The index range is 1-9.

InstantIDConfiguration Class

The InstantIDConfiguration class contains the InstantID credentials.

Class Properties

Name	Type	Description
Password	String	This property contains the InstantID password. This property is required.
UserID	String	This property contains the InstantID user ID. This property is required.

SearchConfiguration Class

The SearchConfiguration class contains the settings to be used in a search.



These settings affect your organization's results. Test these settings thoroughly before including them in your business process.

Class Properties

Name	Type	Description
AssignResultTo	AssignmentInfo	The AssignmentInfo class sets division access and assigns alerts to specified roles or users.
DuplicateMatch Suppression	Boolean	True sets XG to not generate duplicate alerts. The default value is false.
DuplicateMatch SuppressionSameDivision Only	Boolean	True sets XG to not generate duplicate alerts when the duplicate alert and the existing alert can be accessed by the same division. The default value is false.

Class Properties (Continued)

Name	Type	Description
ExcludeScreeningLists	Boolean	True sets XG to exclude list matches for an alert in the response. This option can help reduce network traffic. The response only returns the following properties in the ResultRecord class: • Record • ResultID • RunID • HasScreeningListMatches • RecordStatus If this property is set to false, the ResultRecord class does not return the HasScreeningListMatches and RecordStatus properties. The default value is false.
FraudPoint	FraudPointConfiguration	The FraudPointConfiguration class contains the FraudPoint Score configuration settings. The property performs a FraudPoint Score search. By default, an alert is generated if the score is less than or equal to 630. To configure specific settings, you must create a predefined search in the XG user interface and include the predefined search in the PredefinedSearchName property.
InstantID	InstantIDConfiguration	The InstantIDConfiguration class contains the InstantID credentials. The class performs an InstantID search using the following default settings: • An alert is generated if the index is less than or equal to 30 or if an error is returned. • The DOB (date of birth) match type is Fuzzy match CCYYMMDD. A result DOB must be a close match to the year, month, and day of birth of the full input DOB. To configure specific settings, create a predefined search in the XG user interface and specify that XG use it for the search in the PredefinedSearchName property.
InstantIDIntl	InstantIDIntl Configuration	The LexisNexis® InstantID® International service is no longer available. This class and its properties are not defined.

Class Properties (Continued)

Name	Type	Description
PredefinedSearchName	String	This property contains a predefined search that was created in the XG user interface. Predefined searches provide a specific set of settings that XG uses to perform searches. Automating the selection of these search settings helps you establish more efficient and consistent searching. XG ignores this property if you provide the FraudPoint property, the InstantID property, or the Watchlist property. You must have the division access and the role access that is required to use the predefined search. If search services other than list screening are included in the predefined search, service credentials must be set up through the XG user interface.
Watchlist	WatchlistConfiguration	The WatchlistConfiguration class contains the list search settings.
WriteResultsToDatabase	Boolean	True sets XG to store the search results in the XG database. The default value is false.

Related Information

[FraudPointConfiguration Class](#) on page 31

[InstantIDConfiguration Class](#) on page 33

[WatchlistConfiguration Class](#) on page 41

[AssignmentInfo Class](#) on page 31

[Search Criteria](#)

WatchlistAutomaticFalsePositiveRules Class

The WatchlistAutomaticFalsePositiveRules class contains the standard rules for automatic false positives.



You should not use the standard rules for automatic false positives in combination with rules for LexisNexis® Intelligent Match Decision Solution (IMDS).

You can set the standard rules to determine that a list match in an alert is an automatic false positive when only the input entity name matches a list entity name and the specified data element does not match. You can also set XG to consider a list match in an alert to be an automatic false positive when the input name matches a list entity name and the list match meets the White List criteria.

Application of the Rules

You can set multiple rules for automatic false positives. A list match needs to meet only one of the specified rules for XG to determine that the match is an automatic false positive. However, if any of the following data elements also matches in the same list match, then the product does not set the match as an automatic false positive: address, DOB, ID number, or phone number.

An alert can contain multiple list matches. When only some list matches in the alert meet one of the specified rules, XG generates an alert with an open alert state. By default, this alert contains only the list matches that do not meet a rule. You can configure the product to also include the list matches that meet a rule in the alert.

When all list matches in the alert meet one of the selected rules, XG generates an alert with an open alert state by default. You can configure the product to not generate an alert or to generate an alert with a closed alert state.

Automatic False Positive Example

Your organization decides that matches are false positives when the input entity's gender and the list entity's gender are not the same. You specify the standard rule for gender in the settings that you use to search your input entity.

You search the following input entity: Name: Zach Thul Gender: Female Address: 503 Waybreeze Blvd	XG finds this list entity match: Name: Zach Thul Gender: Male Address: 111 Waybreeze Blvd
--	--

The gender does not match so XG considers the match to be an automatic false positive. However, if the address had matched, XG would not consider the match to be an automatic false positive.

If the alert contains only list matches that meet the Automatic False Positive rule for gender, the predefined search sets XG to complete one of the following actions:

- The product generates an alert with an open alert state.
- The product generates an alert with a closed alert state.
- The product does not generate an alert.

If the alert contains list matches that do not meet the rule, XG generates an alert with an open alert state. The alert contains the list matches that do not meet the rule. The predefined search tells the product whether the alert can also contain the list matches that do meet the rule.



These rules are not applied to searches that are processed with the XG - Unstructured format. You use the XG - Unstructured format to process files with unparsed data that is not provided in defined fields. Because XG cannot distinguish between addresses, IDs, or other data elements in unparsed data, XG searches for only names in these files.

Class Properties

Name	Type	Description
Addresses	Boolean	True sets XG to consider a list match in an alert to be an automatic false positive when the following criteria is met: - The input name matches a list entity name. - The following data elements do not match in the list match: address, DOB, ID number, or phone number. - The input address does not match the address of the list entity. The default value is false.
Citizenship	Boolean	True sets XG to consider a list match in an alert to be an automatic false positive when the following criteria is met: - The input name matches a list entity name. - The following data elements do not match in the list match: address, DOB, ID number, or phone number. - The input citizenship does not match the citizenship of the list entity. The default value is false.
Countries	Boolean	True sets XG to consider a list match in an alert to be an automatic false positive when the following criteria is met: - The input name matches a list entity name. - The following data elements do not match in the list match: address, DOB, ID number, or phone number. - The input country does not match the country of the list entity. Only country names in English, for both the input country and list entity country, are currently supported for this rule to work properly and avoid alerts from being incorrectly suppressed. The default value is false.
DisplayAFPMatches	Boolean	True sets XG to display the list matches that meet at least one of the selected rules when not all of the matches meet a rule. If this option is not selected, XG displays only the matches that do not meet a rule. The default value is false.

Class Properties (Continued)

Name	Type	Description
DOB	Boolean	True sets XG to consider a list match in an alert to be an automatic false positive when the following criteria is met: - The input name matches a list entity name. - The following data elements do not match in the list match: address, DOB, ID number, or phone number. - The input DOB does not match the DOB of the list entity. The default value is <code>false</code>.
DOBTolerance	Integer	This option works with the Date of Birth rule. The option specifies how many months an input DOB can vary from the DOB of a list entity before XG applies the rules for an automatic false positive. For example, the DOB of a list entity is 4/15/1950. If you enter 2 in the Tolerance field, then an input DOB between 2/15/1950 and 6/15/1950 does not activate the Automatic False Positive rules because the DOB is too varied. The default is 0.
EntityType	Boolean	True sets XG to consider a list match in an alert to be an automatic false positive when the following criteria is met: - The input name matches a list entity name. - The following data elements do not match in the list match: address, DOB, ID number, or phone number. - The input entity type does not match the entity type of the list entity. The default value is <code>false</code>.
Gender	Boolean	True sets XG to consider a list match in an alert to be an automatic false positive when the following criteria is met: - The input name matches a list entity name. - The following data elements do not match in the list match: address, DOB, ID number, or phone number. - The input gender does not match the gender of the list entity. The default value is <code>false</code>.
GenerateResultRecord	Boolean	True sets XG to generate an alert when all list matches in the alert meet one of the selected rules. XG generates an alert with a closed alert state. The default value is <code>false</code>.

Class Properties (Continued)

Name	Type	Description
IDs	Boolean	True sets XG to consider a list match in an alert to be an automatic false positive when the following criteria is met: - The input name matches a list entity name. - The following data elements do not match in the list match: address, DOB, ID number, or phone number. - The input ID does not match the ID of the list entity. The default value is false.
Phones	Boolean	True sets XG to consider a list match in an alert to be an automatic false positive when the following criteria is met: - The input name matches a list entity name. - The following data elements do not match in the list match: address, DOB, ID number, or phone number. - The input phone number does not match the phone number of the list entity. The default value is false.
ShowResultAsAlert	Boolean	True sets XG to set an open alert state for alerts that are generated when all list matches in the alert meet at least one of the selected rules.  <i>If this option is not selected, you should consider creating a procedure to process alerts that are set to an Automatic False Positive status, but do not have an open state. This procedure is recommended because you may want to add these alerts to accept lists.</i> The default value is false.
WhiteList	Boolean	True sets XG to consider a list match in an alert to be an automatic false positive when the following criteria is met: - The input name matches a list entity name. - The list match meets the White List criteria. The default value is false.

WatchlistDataFile Class

The WatchlistDataFile class defines the settings for each accept list, custom screening list, or LexisNexis screening list to be searched.

To specify a list, you need the list's exact name or the list's ID number. If you provide both the name and the ID, the ID is used. The name and the ID can be obtained through the following methods:

- The GetDataFileList method
- The GetList method in the List Maintenance interface

Class Properties

Name	Type	Description
Custom	Boolean	True indicates that the list is a custom screening list.
ID	GUID	This property contains a system-generated number that identifies the list. This property is required if the Name property is not provided.
IgnoreWeakAKAs	Boolean	False sets XG to ignore matches on low-quality AKA values. Some source agencies identify low-quality AKA values that are relatively broad and generic as "weak AKAs" in their list records. An example of a weak AKA is "Hassan the Old". The name and descriptor are both common terms. You can choose to exclude low-quality AKA values in a list record from being searched. This option allows you to focus the search on higher-quality entity information. This option may also help reduce the number of false positive matches that XG reports. This functionality is available for the following lists: • OFAC (Office of Foreign Assets Control) SDN (specially designated nationals) list • OFAC Non-SDN Entities list • Ukraine Financial Monitoring list • Some LexisNexis® WorldCompliance™ lists The default value is false.

Class Properties (Continued)

Name	Type	Description
MinScore	Integer	This property contains the minimum match confidence score (50-100) that a match must meet before the match is reported. This score is not a percentage. The score indicates how closely the input entity matched the list entity. A score of 90 indicates the match is closer than a potential match with a score of 80. Your organization's risk tolerance may vary depending on the data matched.  <i>Lower minimum scores may increase the potential matches that are reported. Test different settings to determine the one that best suit your organization's risk tolerance.</i> The default value is 85.
Name	String	This property contains the list name. You can also set the list by selecting the corresponding file check box directly on the Predefined Search page in the XG user interface. This property contains if the ID property is not provided.
SearchCriteria	WatchlistSearchCriteria	The WatchlistSearchCriteria class contains the search criteria that limits the list records that are searched. This functionality is available only for WorldCompliance lists and the UN (United Nations) Consolidated List. For more information about search criteria, see https://bridgerinsight.custhelp.com/app/answers/detail/a_id/882.
TopLevelOperatorIsOr	Boolean	True indicates that the top level operator is OR. The default value is false.

Related Information

[GetDataFileList Method](#) on page 26

[GetList Method](#) on page 124

[WatchlistPaymentScreeningOptions Class](#) on page 50

[Search Criteria](#)

[WatchlistSearchCriteria Class](#) on page 50

WatchlistConfiguration Class

The WatchlistConfiguration class contains the list search settings.

Class Properties

Name	Type	Description
AutomaticFalsePositiveRules	WatchlistAutomaticFalsePositiveRules	The WatchlistAutomaticFalsePositiveRules class contains the standard rules for automatic false positives.
DataFiles	List<WatchlistDataFile>	The WatchlistDataFile class defines the settings for each accept list, custom screening list, or LexisNexis screening list to be searched. This property is required.
FalseMatchSettings	WatchlistFalseMatchSettings	The WatchlistFalseMatchSettings class contains the options for false match updates.
GeneralOptions	WatchlistGeneralOptions	The WatchlistGeneralOptions class contains the general list search options.
IMDS	WatchlistIntelligentMatchDecisionSolution	The WatchlistIntelligentMatchDecisionSolution class contains the options for IMDS.
MatchDispositionRules	WatchlistMatchDispositionRules	The WatchlistMatchDispositionRules class contains the options for match decisions and match updates.
MatchOptions	WatchlistMatchOptions	The WatchlistMatchOptions class contains various search options for list screening.

Related Information

[WatchlistAutomaticFalsePositiveRules Class](#) on page 35

[WatchlistDataFile Class](#) on page 40

[WatchlistGeneralOptions Class](#) on page 44

[WatchlistMatchDispositionRules Class](#) on page 47

[WatchlistMatchOptions Class](#) on page 49

[WatchlistFalseMatchSettings Class](#) on page 43

WatchlistEFTMatchOptions Class

The WatchlistEFTMatchOptions class contains the ACH (Automated Clearing House) options.

You can set how XG handles IAT (International Automated Clearing House Transaction) records.

The IAT structure includes two U.S. OFAC indicators. These indicators are used to communicate that an OFAC match was found for one of the entities that is associated with the transaction by screeners that are external to XG. Typically, XG reports a match if the input IAT record contains either of these indicators. For records that contain both indicators, XG creates only one alert. XG reports the match against the originating entity. If the originating entity is not included in the record, then XG reports the match against the first parsed entity. Along with the match that is reported because of the indicators, XG also reports any additional matches for entities that are associated with the record.

Class Properties

Name	Type	Description
ACHGatewayOperator OFACScreeningIndicator	Boolean	True sets XG to not generate an alert when an input ACH record contains a gateway operator OFAC screening indicator. The default value is false.
ACHSecondaryOFAC ScreeningIndicator	Boolean	True sets XG to not generate an alert when an input ACH record contains a secondary OFAC screening indicator. The default value is false.

WatchlistFalseMatchSettings Class

The WatchlistFalseMatchSettings class contains the options for false match updates.

XG can be set to generate an alert that contains a false match update when new match information is found for certain data elements that are in an accept list record.



For a match that was found on a custom screening list, XG can only generate a false match update if the list is required to contain a unique ID for each record.

False match updates are designed to alert you to new match information that may need further investigation to determine whether the record should remain on the accept list as a false positive. For example, if you previously added a match to an accept list that did not contain a matching DOB, and then XG found a matching DOB in a later search, XG would generate a false match update.

Class Properties

Name	Type	Description
Address	Boolean	True sets XG to return a false match update when a new address match is found for the accept list record. The product only uses this option if the GenerateFalseMatchUpdate property is set to True. The default value is false.
Aka	Boolean	True sets XG to return a false match update when a new AKA match is found for the accept list record. The product only uses this option if the GenerateFalseMatchUpdate property is set to True. The default value is false.
Citizenship	Boolean	True sets XG to return a false match update when a new citizenship match is found for the accept list record. The default value is false.

Class Properties (Continued)

Name	Type	Description
Dob	Boolean	True sets XG to return a false match update when a new DOB match is found for the accept list record. The product only uses this option if the GenerateFalseMatchUpdate property is set to True. The default value is false.
Gender	Boolean	True sets XG to return a false match update when a new gender match is found for the accept list record. The product only uses this option if the GenerateFalseMatchUpdate property is set to True. The default value is false.
GenerateFalseMatch Update	Boolean	True sets XG to return a false match update when a new match on a specified data element is found for the accept list record. The default value is false. True is required for XG to use the other properties in this class.
IdNumbers	Boolean	True sets XG to return a false match update when a new ID match is found for the accept list record. The product only uses this option if the GenerateFalseMatchUpdate property is set to True. The default value is false.
Name	Boolean	True sets XG to return a false match update when a new name match is found for the accept list record. The product only uses this option if the GenerateFalseMatchUpdate property is set to True. The default value is false.
Phone	Boolean	True sets XG to return a false match update when a new phone match is found for the accept list record. The product only uses this option if the GenerateFalseMatchUpdate property is set to True. The default value is false.

Related Information

[WatchlistMatchDispositionRules Class](#) on page 47

WatchlistGeneralOptions Class

The WatchlistGeneralOptions class contains the general list search options.

General options can affect how XG generates an alert, how the match confidence score is determined, and what comparisons are made during the search.

Class Properties

Name	Type	Description
GenerateAlertsForAllRecords	Boolean	True sets XG to generate alerts for all records searched regardless if they generate potential list matches. This property is not available for XG Service.  <i>This option can greatly increase the size of your XG database. This property is ignored if your XG client instance is not configured to generate results for all records checked.</i> The default value is false.
GenerateAlertsForErrors	Boolean	True sets XG to generate an open alert when an error occurs during the search. The default value is false.
IgnoreMatchesAgainstVessels	Boolean	True sets XG to not report name matches for list entities with a vessel entity type. If you do not conduct business with entities that are associated with vessels, you might choose to select this option to help reduce the false positives that are reported. The default value is false.
ScanAddressForBlockedCountries	Boolean	True sets XG to search the Address Line 1 input field and the Address Line 2 input field for certain data in country lists, but only if data is not provided in the City input field, the State input field, or the Country input field. This data includes country names, country AKA names, country terms (for example, Iranian), cities, and ports. This option is designed to find country data in unparsed addresses when the country name is provided in an input address field other than the Country field. For example, you select this option and enter 1 Cuba Street, Anytown, AA 00012 in the Address Line 1 field. You search against a country list that contains Cuba. XG would report a potential match.  <i>If data is provided in any other address fields, XG searches the data as a parsed address, which may return different results.</i> The default value is false.

Class Properties (Continued)

Name	Type	Description
ScanNameForBlockedCountries	Boolean	True sets XG to search name fields for certain data in country lists. This data includes country names, country AKA names, country terms (for example, Iranian), cities, and ports. For example, you perform a search that contains Bank of Iran as the input name. XG would report a potential match against a country list that contains Iran. The default value is false.
ScanRunTogetherWordsForBlockedCountries	Boolean	True sets XG to search input run-together words for country names that are included in country lists. The default value is false.

WatchlistIntelligentMatchDecisionSolution Class

The WatchlistIntelligentMatchDecisionSolution class contains the options for IMDS.



You should not use the standard rules for automatic false positives in combination with rules for IMDS.

IMDS lets your organization develop rules to automatically process list matches in a way that aligns with your organization's list screening program and risk profile. The rules directly utilize input, list, and match attributes from XG searches to test for evidence of a false positive. IMDS uses specified rules and a confidence threshold to determine whether a match is an automatic false positive. Successful outcomes in a well-planned, tuned, and validated implementation of IMDS can result in increased productivity and reduced operational costs.



IMDS is available based on your product subscription. Before you can configure IMDS in a predefined search, the rules must be set up in the system for your XG client instance. For XG Enterprise, see LexisNexis® Bridger Insight® XG Enterprise Manager Help.

If IMDS is configured for your client instance, you can set the rules and the confidence threshold that a match must meet to be considered an automatic false positive. The false positive confidence score indicates how confident the rules are that the match is a false positive. The false positive confidence score can range from 0.0000 (no confidence that the match is a false positive) to 10.0000 (full confidence that the match is a false positive).



The false positive confidence score is not the same as the match confidence score (0-100) that XG provides for each match to indicate how closely the list entity matches the input entity.

For more information about rules for IMDS, see the *Rules for LexisNexis® Intelligent Match Decision Solution* technical reference guide.

Application of the Rules

You can set multiple rules for automatic false positives. A list match needs to meet only one of the specified rules for XG to determine that the match is an automatic false positive. However, if any of the following data elements also matches in the same list match, then the product does not set the match as an automatic false positive: address, DOB, ID number, or phone number.

An alert can contain multiple list matches. When only some list matches in the alert meet one of the specified rules, XG generates an alert with an open alert state. By default, this alert contains only the list matches that do not meet a rule. You can configure the product to also include the list matches that meet a rule in the alert.

When all list matches in the alert meet one of the selected rules, XG generates an alert with an open alert state by default. You can configure the product to not generate an alert or to generate an alert with a closed alert state.

Class Properties

Name	Type	Description
Confidence	String	This property contains the value for the false positive confidence score that a match must meet to be considered an automatic false positive. Possible values range between 0.0000–10.0000.
RulesName	String	This property contains the rules for IMDS that XG uses to determine that a match is an automatic false positive.

WatchlistMatchDispositionRules Class

The WatchlistMatchDispositionRules class contains the options for match decisions and match updates.

Match decisions are used to separately resolve each match in an alert. Match updates notify you of changes to previously reviewed matches that were added to an accept list.

Class Properties

Name	Type	Description
CheckForFalseMatch Updates	Boolean	True sets XG to generate a false match update when new best name matches or new DOB matches are found for the matches in the accept list record. This deprecated property from XG 4 was replaced by the WatchlistFalseMatchSettings class. LexisNexis recommends that you use the WatchlistFalseMatchSettings class, which lets you select more data elements to generate false match updates.  <i>If the search request includes the WatchlistFalseMatchSettings class and the CheckForFalseMatchUpdates property in the WatchlistMatchDispositionRules class, the product uses the WatchlistFalseMatchSettings class settings.</i> The default value is false.
GenerateTrueMatch Updates	Boolean	True to generate an alert that contains a true match update when list entity information changes for a true match in an accept list record.  <i>XG does not generate a true match update when changes occur only to the following profile information: the entity unique ID, the reference ID, the date that the profile was listed, or the date that the profile was modified.</i> The default value is false.
IndividualMatch Disposition	Boolean	True sets XG to display options for a user to indicate whether an alert is a false positive or a true match. The default value is false.
TrueMatchSetsOthersTo FalsePositive	Boolean	True sets XG to set other WorldCompliance matches to False Positive when a WorldCompliance match is set to True Match. The default value is false.

Related Information

[WatchlistFalseMatchSettings Class](#) on page 43

WatchlistMatchOptions Class

The WatchlistMatchOptions class contains various search options for list screening.

The match options set XG to use addresses, IDs, and phone numbers in addition to name to find candidate records in parsed data. Match options are not applied to searches that are processed using the XG - Unstructured format. For consistent results, you should include the properties that are affected by your selected match options in your requests.



These options increase the scope of your search and can decrease the speed of the search.

Class Properties

Name	Type	Description
Addresses	Boolean	True sets XG to also use the address input fields to find candidate records. The default value is false.
IDs	Boolean	True sets XG to also use the ID input fields to find candidate records. The default value is true.
InfluencedByInitials	Boolean	True sets XG to increase the score when the first name or the middle name is an initial. For example, you search your input entity Imaginerry T Person against a custom list that contains Imaginary Test Person. XG returns a score of 88 if this option is not selected. XG returns a score of 92 if this option is selected. This option may also increase the score if an alternative name for the input name matches a first-name initial or a middle-name initial in the entity name on the list. For example, you search your input entity Patricia Test Person against a custom list that contains T Test Person. XG returns a higher score if this option is selected because the alternative names for Patricia (Trisha or Trish) matched T in the entity name on the custom list. The default value is false.
InfluencedBySingleWords	Boolean	True sets XG to increase the score for matches on single-word organization names. For example, this option may increase scores for matches on acronyms. The default value is false.
Phones	Boolean	True sets XG to also use the phone input fields to find candidate records. The default value is true.

Related Information

[WatchlistPaymentScreeningOptions Class](#) on page 50

WatchlistPaymentScreeningOptions Class

The WatchlistPaymentScreeningOptions class defines the payment transaction options that are available for list screening.

Payment screening options let you configure XG to search each field independently in payment data records for certain data element matches.

Class Properties

Name	Type	Description
EFTOptions	WatchlistEFTMatchOptions	The WatchlistEFTMatchOptions class contains the ACH options.
ScanAllFieldsForBICs	Boolean	True sets the LexisNexis® Search Core to search input fields for a SWIFT (Society for Worldwide Interbank Financial Telecommunication) BIC (Business Identifier Code). The default value is false.
ScanAllFieldsForCountryCodes	Boolean	True sets the LexisNexis Search Core to search input fields for country codes. Two-character and three-character country codes from ISO (International Organization for Standardization) are supported. This property is not used if the ScanAllFieldsForCountryNames is set to false. The default value is false.
ScanAllFieldsForCountryNames	Boolean	True sets the LexisNexis Search Core to search input fields for country names. You can scan for country codes only if you scan for country names. The default value is false.
ScanAllFieldsForNames	Boolean	True sets the LexisNexis Search Core to search input fields for entity names. The default value is false.

Related Information

[WatchlistEFTMatchOptions Class](#) on page 42

WatchlistSearchCriteria Class

The WatchlistSearchCriteria class contains the search criteria that limits the list records that are searched. This functionality is available only for WorldCompliance lists and the UN Consolidated List.

For more information about search criteria, see https://bridgerinsight.custhelp.com/app/answers/detail/a_id/882.

Class Properties

Name	Type	Description
Groups	List<WatchlistSearchCriteriaGroup>	This property contains the groups in which there are search criteria values that you want to use to limit the list records that are searched.

Related Information

[WatchlistSearchCriteriaGroup Class](#) on page 51

[Search Criteria](#)

WatchlistSearchCriteriaGroup Class

The WatchlistSearchCriteriaGroup class contains the group ID and the search criteria values.

For more information about search criteria, see https://bridgerinsight.custhelp.com/app/answers/detail/a_id/882.

Class Properties

Name	Type	Description
Criteria	List<Integer>	This property contains the ID for each search criteria value in the selected group that you want to use to limit the list records that are searched.
ID	Integer	This property contains the ID for the search criteria group.

Related Information

[Search Criteria](#)

Search Input Classes

The search request includes the classes that contain the input entity information.

Related Information

[Search Request Classes](#) on page 29

InputBasicID Class

The InputBasicID class contains the input ID. This property is typically used to search screening list data for a specific proprietary ID for the entity that is provided by the source agency or by LexisNexis. Matches to this search are not included in alerts and are not stored in the result database.

Class Properties

Name	Type	Description
Number	String	This property contains the ID number.
Type	IDType	This property contains the ID type. The ID type is typically ProprietaryUID. This property is required.

Related Information

[Possible ID Types](#) on page 180

InputEFT Class

The InputEFT class contains input payment transaction information or input unstructured information. This property is used to search unstructured data or payment transaction data.

XG does not return the input information that was searched using the Unstructured EFT (electronic funds transfer) type.

Class Properties

Name	Type	Description
Type	EFTType	This property contains the payment transaction type. Possible values: - None - ACH - Fedwire - ISO20022 - SWIFT - Unstructured None is not valid in a request. XML in ISO20022 requests must be encoded before processing.
Value	String	This property contains the input payment transaction record or the input unstructured record.

Related Information

[Searchable Data](#) on page 10

InputEntity Class

The InputEntity class contains the structured input record for business entities, individual entities, or unknown entities. This property is used to search data that is parsed into a structured format.

Class Properties

Name	Type	Description
AdditionalInfo	List<InputAdditionalInfo>	The InputAdditionalInfo class contains the additional information that helps identify the input entity.
Addresses	List<InputAddress>	The InputAddress class contains the input address information. An address is required for InstantID business searches.
EntityType	InputEntityType	This property contains the entity type. Possible values: • None • Business • Individual • Unknown None is not valid in a request.  <i>Records with an unknown entity type can only be searched against screening lists.</i> This property is required.
Gender	GenderType	This property contains the entity gender type. • None • Female • Male • Unknown None is not valid in a request. You use this property only for individual entities.
IDs	List<InputID>	The InputID class contains the input ID information.
Name	InputName	The InputName class contains the input name information. This property is required.
Phones	List<InputPhone>	The InputPhone class contains the input phone number information.

Related Information

[InputAdditionalInfo Class](#) on page 169

[InputAddress Class](#) on page 169

[InputID Class](#) on page 170

[InputName Class](#) on page 172

[InputPhone Class](#) on page 172

InputRecord Class

The InputRecord class contains an input record to be searched. This class is required for all Search requests.

You must provide one of the following properties: **InputEntity**, **InputText**, **InputEFT**, or **InputBasicID**. You should only provide the one of these properties. If you do provide more than one, XG only searches one of them in the following priority:

1. **InputEntity**
2. **InputText**
3. **InputEFT**
4. **InputBasicID**

Class Properties

Name	Type	Description
EFT	InputEFT	The InputEFT class contains input payment transaction information or input unstructured information. This property is used to search unstructured data or payment transaction data.
Entity	InputEntity	The InputEntity class contains the structured input record for business entities, individual entities, or unknown entities. This property is used to search data that is parsed into a structured format.
ID	InputBasicID	The InputBasicID class contains the input ID. This property is typically used to search screening list data for a specific proprietary ID for the entity that is provided by the source agency or by LexisNexis.
RecordID	Long	This property contains the user-defined number that identifies the record in the request. The same number is returned in the Record property of the ResultRecord class. For payment data searches and unstructured data searches, XG does not return the input information when no list matches are reported. You can use this number to track the record.
Text	InputText	The InputText class contains the input information for a text record. This property is used to search unstructured data that also contains a defined ID.

Related Information

- [Searchable Data](#) on page 10
- [InputBasicID Class](#) on page 52
- [ResultRecord Class](#) on page 62
- [InputEFT Class](#) on page 52
- [InputEntity Class](#) on page 53
- [InputText Class](#) on page 55

InputText Class

The InputText class contains the input information for a text record. This property is used to search unstructured data that also contains a defined ID.

Class Properties

Name	Type	Description
IDs	InputID	The InputID class contains the input ID information.
Text	String	This property contains the input text string.

Related Information

[InputID Class](#) on page 170

SearchInput Class

The SearchInput class contains the input records. For a search that includes entity records or text records, the request can contain a maximum of 100 records. For a search that contains payment transaction records, the request can contain multiple records with a total maximum size of 1 MB for the entire search input. This class is required for all Search requests.

Class Properties

Name	Type	Description
BlockID	String	This property contains the block ID.
Records	List<InputRecord>	The InputRecord class contains an input record to be searched. This property is required.

Related Information

[InputRecord Class](#) on page 54

Search Response Classes

XG returns Search interface responses as objects.

This information is defined by the following classes:

- "Record Information Classes" on page 60
- "FraudPoint Score Result Classes" on page 67
- "InstantID Result Classes" on page 70
- "List Screening Result Classes" on page 79

Depending on the method, many of these classes can also be included in the request or in the response when managing results in the Results interface or managing list records in the List Maintenance interface.

Some classes contain other classes as properties. The classes are described in alphabetical order. Classes that are used in multiple web services interfaces may be described in another section.

Related Information

[Result Management](#) on page 99

[List Management](#) on page 122

Search Response Class Hierarchy

A class hierarchy depicts the structure of the classes within the code.

The following class hierarchy depicts the structure of the search response:

"ClientContext Class" on page 21

"SearchResults Class" on page 67 > "ResultRecord Class" on page 62:

- "FraudPointResults Class" on page 69:
 - "FPInput Class" on page 68:
 - "FPAddress Class" on page 68
 - "InputName Class" on page 172
 - "FPRedFlag Class" on page 68 > "FPRiskIndicator Class" on page 69
 - "FPRiskIndicator Class" on page 69
 - "ResultError Class" on page 187
- "InstantIDBusinessResults Class" on page 76:
 - "IIDFEINMatch Class" on page 72 > "IIDAddress Class" on page 71
 - "IIDFiling Class" on page 72 > "IIDAddress Class" on page 71
 - "IIDInput Class" on page 73:
 - "IIDAddress Class" on page 71
 - "InputName Class" on page 172
 - "IIDReversePhone Class" on page 74:
 - "IIDAddress Class" on page 71
 - "InputName Class" on page 172
 - "IIDRiskIndicator Class" on page 74
 - "IIDVerificationIndicators Class" on page 75
 - "IIDWatchlist Class" on page 75 > "IIDAddress Class" on page 71
 - "ResultError Class" on page 187
- "InstantIDIndividualResults Class" on page 77:
 - "IIDAddressHistory Class" on page 71 > "IIDAddress Class" on page 71
 - "IIDAlternateName Class" on page 71
 - "IIDInput Class" on page 73:
 - "IIDAddress Class" on page 71
 - "InputName Class" on page 172
 - "IIDRedFlag Class" on page 73 > "IIDRiskIndicator Class" on page 74
 - "IIDReversePhone Class" on page 74:
 - "IIDAddress Class" on page 71
 - "InputName Class" on page 172
 - "IIDRiskIndicator Class" on page 74
 - "IIDSSNInfo Class" on page 74
 - "IIDWatchlist Class" on page 75 > "IIDAddress Class" on page 71

- "ResultError Class" on page 187
- "ResultRecordDetails Class" on page 63:
 - "InputAdditionalInfo Class" on page 169 > "Date Class" on page 95
 - "InputAddress Class" on page 169
 - "InputID Class" on page 170 > "Date Class" on page 95
 - "InputName Class" on page 172
- "ParsedACH Class" on page 148:
 - "ParsedACHField Class" on page 150
 - "ParsedACHNote Class" on page 150
 - "ParsedACHParty Class" on page 151:
 - "ParsedACHAdditionalInfo Class" on page 149
 - "ParsedACHAddress Class" on page 149
 - "ParsedACHID Class" on page 150
- "ParsedFedwire Class" on page 154:
 - "ParsedFedwireField Class" on page 156 > "ParsedFedwireAmount Class" on page 155
 - "ParsedFedwireNote Class" on page 156
 - "ParsedFedwireParty Class" on page 157:
 - "ParsedFedwireAdditionalInfo Class" on page 154
 - "ParsedFedwireAddress Class" on page 154
 - "ParsedFedwireNote Class" on page 156
 - "ParsedFedwireID Class" on page 156
 - "ParsedFedwirePhone Class" on page 158
- "ParsedISO20022 Class" on page 158:
 - "ParsedISO20022Field Class" on page 160
 - "ParsedISO20022Note Class" on page 161
 - "ParsedISO20022Party Class" on page 161:
 - "ParsedISO20022AdditionalInfo Class" on page 159
 - "ParsedISO20022Address Class" on page 159
 - "ParsedISO20022ID Class" on page 160
 - "ParsedISO20022Note Class" on page 161
 - "ParsedISO20022Phone Class" on page 162
- "ParsedSWIFT Class" on page 163:
 - "ParsedSWIFTField Class" on page 165:
 - "ParsedSWIFTAmount Class" on page 165

- "ParsedSWIFTOriginalMessage Class" on page 166
- "ParsedSWIFTNote Class" on page 166
- "ParsedSWIFTParty Class" on page 167:
 - "ParsedSWIFTAdditionalInfo Class" on page 164
 - "ParsedSWIFTAddress Class" on page 164
 - "ParsedSWIFTID Class" on page 166
- "InputPhone Class" on page 172
- "RecordState Class" on page 61:
 - "AuditItem Class" on page 60
 - "WatchlistMatchState Class" on page 67
- "ResultError Class" on page 187
- "WatchlistResults Class" on page 80:
 - "OFACReportInfo Class" on page 80 > "OFACInstitutionInfo Class" on page 79
 - "ResultError Class" on page 187
 - "WLAddressMatch Class" on page 81
 - "WLCitizenshipMatch Class" on page 81
 - "WLCountryDetails Class" on page 82
 - "WLDOBMatch Class" on page 82
 - "WLEntityDetails Class" on page 83:
 - "EntityAdditionalInfo Class" on page 174
 - "EntityAddress Class" on page 174
 - "EntityAKA Class" on page 175
 - "EntityID Class" on page 176 > "Date Class" on page 95
 - "EntityName Class" on page 176
 - "EntityPhone Class" on page 177
- "WLMatch Class" on page 84:
 - "WLIDMatch Class" on page 84
 - "WLMatchConflicts Class" on page 88
 - "WLMatchFile Class" on page 88
 - "WLMatchFprDetails Class" on page 89
 - "WLMatchSummary Class" on page 89 > "WLMatchSummaryItem Class" on page 90
 - "WLPhoneMatch Class" on page 90

Record Information Classes

The search response includes the classes that contain information about the record details, the record state, and the record history.

Related Information

[Search Response Class Hierarchy](#) on page 56

AuditItem Class

The AuditItem class contains the events that are associated with an alert. For auditing purposes, this record history can provide an account of your organization's processing actions on the entity.

Class Properties

Name	Type	Description
Date	DateTime	This property contains the date (YYYY-MM-DD) and the 24-hour local time (HH:MM:SS) that the event was added to the record history.
Event	String	This property contains the event type. Possible values for XG 5: • Added to Accept List • Added to Custom Screening List • Alert Decision Applied • Alert Opened or Closed • Archive Transfer • Attachment Added or Attachment Deleted • Decision Tags Changed • LexisNexis FraudPoint • LexisNexis InstantID • LexisNexis InstantID International • LexisNexis News • List Screening Search Conducted • New Note • OFAC Report Generated • Reassigned • Record Created • Record Details Changed • Record Rerun • Status Changed • Field Truncated

Class Properties (Continued)

Name	Type	Description
Note	String	For a Note Added event, this property contains the note that was provided by the user. For other events, this property contains system information about the event.
User	String	This property contains the user name of the person who performed the event.

RecordState Class

The RecordState class contains the alert state information.

Class Properties

Name	Type	Description
AddedToAcceptList	Boolean	True indicates that the alert was added to an accept list.
AlertState	AlertStateType	This property contains the alert state. Possible values: - None - Closed - Open - Unassigned None is not valid in a request. Unassigned is only returned for alerts that did not contain results. For example, this value is set if the search generated an error but not an alert.
AssignedTo	List<String>	This property contains the user or the list of roles. You cannot assign the alert to a user or role if the state is set to Closed. Only users who belong to the division can process the alert.
AssignmentType	AssignmentType	This property contains the assignment type. Possible values: - Role - Unknown - User
Division	String	This property contains the division that can access the alert. Only users who belong to the division can process the alert.
History	List<AuditItem>	The AuditItem class contains the events that are associated with an alert. This property is only returned as part of search results.

Class Properties (Continued)

Name	Type	Description
MatchStates	List<WatchlistMatchState>	The WatchlistMatchState class contains the decision information if separate decisions were set for each list match in the alert.
Note	String	This property contains user-defined information.
Status	String	This property contains the alert status. The status value must correspond with an alert status set up in the XG application. The values are not case sensitive and will reflect the default case used in the XG application. To avoid possible errors and clearly indicate what stage the alert is at in your result process, XG can be configured to require an alert status when alerts are closed or reassigned. If a required alert status is not set, an error is returned.

Related Information

[AuditItem Class](#) on page 60

[WatchlistMatchState Class](#) on page 67

ResultRecord Class

The ResultRecord class contains the alert that XG generated for the search.

When you add the alert to an accept list or a custom screening list, XG ignores the information that is not relevant to list searching. For example, results from search services other than list screening are not added to an accept list or a custom screening list.

Class Properties

Name	Type	Description
AttachmentIDs	List<Long>	This property contains the system-generated number that identifies a file attached to the alert.
FraudPoint	FraudPointResults	The FraudPointResults class contains the FraudPoint Score results.
HasScreeningListMatches	Boolean	True indicates that the alert contains list matches. This property is returned only if ExcludeScreeningListMatches parameter is set to true in the request to exclude list matches in the response.
InstantIDBusiness	InstantIDBusinessResults	The InstantIDBusinessResults class contains the results for InstantID business searches.
InstantIDIndividual	InstantIDIndividualResults	The InstantIDIndividualResults class contains the results for InstantID individual searches.
InstantIDIntlIndividual	InstantIDIntlIndividualResults	The InstantID International service is no longer available. This class and its properties are not defined.

Class Properties (Continued)

Name	Type	Description
Record	Long	This property contains the user-defined number that identifies the alert in the request.
RecordDetails	ResultRecordDetails	The ResultRecordDetails class contains the account information, the input data, and the other details about the alert. This property is required when you add the alert to an accept list or custom screening list.
RecordStatus	String	This property contains the current status of the alert. This property is returned only if the ExcludeScreeningListMatches parameter is set to true in the request to exclude list matches in the response.
ResultID	Long	This property contains the system-generated number that identifies the alert. This property is required when you add the alert to an accept list.
RunID	Long	This property contains the system-generated number that identifies the search if the alerts were stored in the XG database. This property is required when you add the alert to an accept list.
Watchlist	WatchlistResults	The WatchlistResults class indicates the status of the list search and contains potential matches. This property is required when you add the alert to an accept list.

Related Information

[FraudPointResults Class](#) on page 69
[InstantIDBusinessResults Class](#) on page 76
[InstantIDIndividualResults Class](#) on page 77
[InstantIDIntlIndividualResults Class](#)
[ResultRecordDetails Class](#) on page 63
[WatchlistResults Class](#) on page 80

ResultRecordDetails Class

The ResultRecordDetails class contains the account information, the input data, and the other details about the alert.

Class Properties

Name	Type	Description
AcceptListID	Long	This property contains the system-generated number that identifies the accept list to which the record was added.
AccountAmount	String	This property contains the account amount.
AccountDate	String	This property contains the account date.
AccountGroupID	String	This property contains the account group ID.
AccountOtherData	String	This property contains the user-defined account data.
AccountProviderID	String	This property contains the account provider ID.
AccountType	String	This property contains the account type.
AddedToAcceptList	Boolean	True indicates that the alert was added to an accept list.
AdditionalInfo	List<InputAdditionalInfo>	The InputAdditionalInfo class contains the additional information that helps identify the input entity.
Addresses	List<InputAddress>	The InputAddress class contains the input address information. If you add the record to an accept list, this property or the IDs property is required.
CustomWatchlistIDs	List<Integer>	This property contains the system-generated number that identifies the custom screening list to which the record was added.
Division	String	This property contains the division that can access the alert. Only users who belong to the division can process the alert.
DPPA	DPPAChoiceType	This property contains the DPPA (Driver's Privacy Protection Act) permissible use. Possible values: • NA • Choice0 • Choice1 • Choice2 • Choice3 • Choice4 • Choice6 The default value is NA. This value is not valid if a DPPA permissible use is required. This property is required for FraudPoint Score searches or InstantID searches on individual entities.
EFTContext	String	If a list match is generated, the input payment transaction information in which the match was found.

Class Properties (Continued)

Name	Type	Description
EFTType	EFTType	This property contains the payment transaction type. Possible values: • None • ACH • Fedwire • ISO20022 • SWIFT • Unstructured
EntityType	ResultEntityType	This property contains the entity type. Possible values: • Business • Individual • Text • Unknown • Unstructured  <i>Only business entities and individual entities can be added to accept lists or custom screening lists.</i>
Error	ResultError	The ResultError class returns any errors that were generated during a search.
Gender	GenderType	This property contains the entity gender type. Possible values: • None • Female • Male • Unknown None is not valid in a request.
GLB	Integer	This property contains the GLBA (Gramm-Leach-Bliley Act) permissible use. Possible values: • NA • 0 • 1 • 3 • 5 • 6 • 7 • 12 The default value is NA. This value is not valid if a GLBA permissible use is required. This property is required for FraudPoint Score searches or InstantID searches.

Class Properties (Continued)

Name	Type	Description
IDs	List<InputID>	The InputID class contains the input ID information. If you add the record to an accept list, this property or the Addresses property is required.
LastUpdatedDate	DateTime	This property contains the date (YYYY-MM-DD) and the 24-hour local time (HH:MM:SS) that the alert was last updated.
Name	InputName	The InputName class contains the input name information. This property is required if you add the record to an accept list or custom screening list.
ParsedACH	ParsedACH	The ParsedACH class contains the input information.
ParsedFedwire	ParsedFedwire	The ParsedFedwire class contains the Fedwire input information.
ParsedISO20022	ParsedISO20022	The ParsedISO20022 class contains the ISO 20022 input information.
ParsedSWIFT	ParsedSWIFT	The ParsedSWIFT class contains the SWIFT input information.
Phones	List<InputPhone>	The InputPhone class contains the input phone number information.
RecordState	RecordState	The RecordState class contains the alert state information.
SearchDate	DateTime	This property contains the date (YYYY-MM-DD) and the 24-hour local time (HH:MM:SS) that XG performed the search.
Text	String	This property contains the text string for an input text record.

Related Information

[InputAdditionalInfo Class](#) on page 169

[InputAddress Class](#) on page 169

[ResultError Class](#) on page 187

[InputID Class](#) on page 170

[InputName Class](#) on page 172

[ParsedACH Class](#) on page 148

[ParsedFedwire Class](#) on page 154

[ParsedISO20022 Class](#) on page 158

[ParsedSWIFT Class](#) on page 163

[InputPhone Class](#) on page 172

[RecordState Class](#) on page 61

SearchResults Class

The SearchResults class contains the search results.

Class Properties

Name	Type	Description
BlockID	String	This property contains the block ID.
ClientReference	String	This property contains user-defined information to track the search.
Records	List<ResultRecord>	The ResultRecord class contains the alert that XG generated for the search.
SearchEngineVersion	String	This property contains the XG version that was used to perform the search (for example, 5.0.0.1764).

Related Information

[ResultRecord Class](#) on page 62

WatchlistMatchState Class

The WatchlistMatchState class contains the decision information if separate decisions were set for each list match in the alert.

Class Properties

Name	Type	Description
MatchID	MatchID	This property contains the system-generated number that identifies the list match.
Type	WatchlistMatchStateType	This property contains the list match state. Possible values: - FalsePositive - TrueMatch

Related Information

[WatchlistMatchDispositionRules Class](#) on page 47

FraudPoint Score Result Classes

The search response includes the classes that contain FraudPoint Score results.

Related Information

[Search Response Class Hierarchy](#) on page 56

FPAddress Class

The FPAddress class contains the input address information.

Class Properties

Name	Type	Description
City	String	This property contains the city.
PostalCode	String	This property contains the postal code.
State	String	This property contains the state.
Street1	String	This property contains the street address.

FPInput Class

The FPInput class contains the input information.

Class Properties

Name	Type	Description
Address	FPAddress	The FPAddress class contains the input address information.
DOB	String	This property contains the DOB.
Name	InputName	The InputName class contains the input name information.
Phone	String	This property contains the phone number.
SSN	String	This property contains the SSN (Social Security number).

Related Information

[FPAddress Class](#) on page 68

[InputName Class](#) on page 172

FPRedFlag Class

The FPRedFlag class contains the red flag details.

For possible red flags, see the *LexisNexis® FraudPoint® Score Analyzing Results* white paper on the LexisNexis Technical Support website at: https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1441.

Class Properties

Name	Type	Description
Name	String	This property contains the red flag category name.
RiskIndicators	List<FPRiskIndicator>	The FPRiskIndicator class contains the red flags details.

Related Information

[FPRiskIndicator Class](#) on page 69

FPRiskIndicator Class

The FPRiskIndicator class contains the risk indicator details or the red flag details.

For possible risk indicators or red flags, see the *LexisNexis® FraudPoint® Score Analyzing Results* white paper on the LexisNexis Technical Support website at: https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1441.

Class Properties

Name	Type	Description
Description	String	This property contains the description of the risk indicator or the red flag.
RiskCode	String	This property contains the risk code that is associated with the risk indicator or the red flag.

FraudPointResults Class

The FraudPointResults class contains the FraudPoint Score results.

For information about the score that is returned, see the *LexisNexis® FraudPoint® Score Analyzing Results* white paper on the LexisNexis Technical Support website at: https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1441.

Class Properties

Name	Type	Description
Errors	List<ResultError>	The ResultError class returns any errors that were generated during a search.
FriendlyFraudIndex	Integer	The index indicates the identity profile is at risk for being a victim of identity fraud perpetrated by a close associate. For example, the individual has high-risk relatives and close associates. The index range is 1-9.
Input	FPIInput	The FPIInput class contains the input information.

Class Properties (Continued)

Name	Type	Description
ManipulatedIdentityIndex	Integer	The index indicates the risk that the identity is being intentionally manipulated. For example, there is a suspicious amount of variation of the PII associated with this identity. The index range is 1-9.
RedFlags	List<FPRedFlag>	The FPRedFlag class contains the red flag details.
RiskIndicators	List<FPRiskIndicator>	The FPRiskIndicator class contains the risk indicator details.
Score	Integer	This property contains the score that indicates a fraud potential.
StolenIdentityIndex	Integer	The index indicates the risk that the transaction is in connection with a real identity that has been compromised. For example, the identity is applying with a new address that is not likely associated with the true owner of the identity. The index range is 1-9.
SuspiciousActivityIndex	Integer	The index indicates the level of suspicious and high-risk activity that was previously observed for the input elements. For example, the input elements are tied to records that were previously deemed to be suspicious or high risk. The index range is 1-9.
SyntheticIdentityIndex	Integer	The index indicates the risk that the input identity is a synthetic identity. For example, this identity is only reported by credit bureaus or is missing characteristics of normal identities. The index range is 1-9.
VulnerableVictimIndex	Integer	The index indicates the identity profile is at risk for being a victim of identity fraud. For example, a high-occupancy location or minors. The index range is 1-9.

Related Information

[ResultError Class](#) on page 187

[FPInput Class](#) on page 68

[FPRedFlag Class](#) on page 68

[FPRiskIndicator Class](#) on page 69

InstantID Result Classes

The search response includes the classes that contain InstantID results.

Related Information

[Search Response Class Hierarchy](#) on page 56

IIDAddress Class

The IIDAddress class contains the input address information and the other addresses InstantID found were associated with the input entity.

Class Properties

Name	Type	Description
City	String	This property contains the city.
County	String	This property contains the county.
PostalCode	String	This property contains the postal code.
State	String	This property contains the state.
Street1	String	This property contains the street address.

IIDAddressHistory Class

The IIDAddressHistory class contains the other addresses associated with the input entity.

Class Properties

Name	Type	Description
Address	IIDAddress	The IIDAddress class contains the input address information and the other addresses InstantID found were associated with the input entity.
DateFirstSeen	String	This property contains the date that InstantID first found the address in public records.
DateLastSeen	String	This property contains the date that InstantID last found the address in public records.
IsBestAddress	String	This property indicates whether the address is the most likely current address of the input entity.

Related Information

[IIDAddress Class](#) on page 71

IIDAlternateName Class

The IIDAlternateName class contains an alternative last name that is associated with the InstantID input entity.

Class Properties

Name	Type	Description
LastName	String	This property contains the alternative last name.
LastSeenDate	String	This property contains the date that InstantID last found the alternative last name in public records.

IIDFEINMatch Class

The IIDFEINMatch class contains the company name and address that are associated with the input FEIN (federal employer identification number).

Class Properties

Name	Type	Description
Address	IIDAddress	The IIDAddress class contains the input address information and the other addresses InstantID found were associated with the input entity.
CompanyName	String	This property contains the company name.

Related Information

[IIDAddress Class](#) on page 71

IIDFiling Class

The IIDFiling class contains the civil filing information that is returned for the input business.

Class Properties

Name	Type	Description
Address	IIDAddress	The IIDAddress class contains the input address information and the other addresses InstantID found were associated with the input entity.
Date	String	This property contains the date of the civil filing.
Name	String	This property contains the name on the civil filing.
Type	String	This property contains the type of civil filing.

Related Information

[IIDAddress Class](#) on page 71

IIDInput Class

The IIDInput class contains the input information and the verified information.

Class Properties

Name	Type	Description
Address	IIDAddress	The IIDAddress class contains the input address information and the other addresses InstantID found were associated with the input entity.
DOB	String	This property contains the DOB.
DOBVerified	String	This property indicates whether the DOB is verified. Possible values: 0 = False 1 = True
FEIN	String	This property contains the FEIN.
Name	InputName	The InputName class contains the input name information.
Phone	String	This property contains the phone number.
SSN	String	This property contains the SSN.

Related Information

[IIDAddress Class](#) on page 71

[InputName Class](#) on page 172

IIDRedFlag Class

The IIDRedFlag class contains the red flags.

Class Properties

Name	Type	Description
Name	String	This property contains the red flag category.
RiskIndicators	List<IIDRiskIndicator>	The IIDRiskIndicator class contains the red flags.

Related Information

[IIDRiskIndicator Class](#) on page 74

IIDReversePhone Class

The IIDReversePhone class contains the name and the address that are associated with the input phone number.

Class Properties

Name	Type	Description
Address	IIDAddress	The IIDAddress class contains the input address information and the other addresses InstantID found were associated with the input entity.
Name	InputName	The InputName class contains the input name information.

Related Information

[IIDAddress Class](#) on page 71

[InputName Class](#) on page 172

IIDRiskIndicator Class

The IIDRiskIndicator class contains the risk indicators and the red flags. For individual searches, this class also returns the NAP verification value and the NAS verification value.

For more information about InstantID results, see the *LexisNexis® InstantID® Analyzing Results* white paper on the LexisNexis Technical Support website at https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1442.

Class Properties

Name	Type	Description
Description	String	This property contains the description of the risk indicator.
RiskCode	String	This property contains the risk code that is associated with the risk indicator or the red flag.

IIDSSNInfo Class

The IIDSSNInfo class contains the SSN result information.

Class Properties

Name	Type	Description
IssuedEndDate	String	This property contains the end date of the time period that the SSN was issued.

Class Properties (Continued)

Name	Type	Description
IssuedLocation	String	This property contains the location where the SSN was issued.
IssuedStartDate	String	This property contains the start date of the time period that the SSN was issued.

IIDVerificationIndicators Class

The IIDVerificationIndicators class contains the verification information that is returned for InstantID business searches.

Class Properties

Name	Type	Description
Address	String	This property contains the street address.
City	String	This property contains the city.
FEIN	String	This property contains the FEIN.
Name	String	This property contains the business name.
Phone	String	This property contains the phone number.
PostalCode	String	This property contains the postal code.
State	String	This property contains the state.

IIDWatchlist Class

The IIDWatchlist class contains the list information.



The IIDWatchlist class is never returned because XG provides its own list screening.

Class Properties

Name	Type	Description
Address	IIDAddress	The IIDAddress class contains the input address information and the other addresses InstantID found were associated with the input entity.
County	String	This property contains the county.
EntityName	String	This property contains the entity name.
FirstName	String	This property contains the first name.
LastName	String	This property contains the last name.
RecordNumber	String	This property contains the record number.
Table	String	This property contains the table.

Related Information

[IIDAddress Class](#) on page 71

InstantIDBusinessResults Class

The InstantIDBusinessResults class contains the results for InstantID business searches.

For possible results, see the *LexisNexis® InstantID® Analyzing Results* white paper on the LexisNexis Technical Support website at: https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1442.

Class Properties

Name	Type	Description
BankruptcyFilingsCount	Integer	This property contains the number of bankruptcy filings.
ComprehensiveVerificationIndex	Integer	This property contains the overall risk index (0, 10, 20, 30, 40, or 50).
CorrectedCompanyName	String	This property contains the corrected company name.
Errors	List<ResultError>	The ResultError class returns any errors that were generated during a search.
FEINMatches	List<IIDFEINMatch>	The IIDFEINMatch class contains the company name and address that are associated with the input FEIN.
Input	IIDInput	The IIDInput class contains the input information and the verified information.
MostRecentBankruptcyFiling	IIDFiling	The IIDFiling class contains the civil filing information that is returned for the input business.
MostRecentJudgmentLienFiling	IIDFiling	The IIDFiling class contains the civil filing information that is returned for the input business.
NameAddressFEINIndicator	Integer	This property contains the BNAF (business name-address-FEIN) verification index that is used to calculate the overall risk index that is returned in the ComprehensiveVerificationIndex property.
NameAddressPhoneIndicator	Integer	This property contains the business name-address-phone verification index that is used to calculate the overall risk index that is returned in the ComprehensiveVerificationIndex property.
NameAddressSSNIndicator	Integer	This property contains the BNAS (business name-address-SSN) verification index that is used to calculate the overall risk index that is returned in the ComprehensiveVerificationIndex property.
PhoneOfNameAddress	String	This property contains the phone number that is related to the input name and the input address.

Class Properties (Continued)

Name	Type	Description
PhoneType	String	This property contains the phone type. Possible values: • B = Blank Input • 0 = Residential • 2 = Business • 7 = Mobile/Cellular • 8 = Pager
ReleasedJudgmentLienCount	Integer	This property contains the number of released judgments.
ReversePhone	IIDReversePhone	The IIDReversePhone class contains the name and the address that are associated with the input phone number.
RiskIndicators	List<IIDRiskIndicator>	This property contains the IIDRiskIndicator class contains the risk indicators.
UnreleasedJudgmentLienCount	Integer	This property contains the number of unreleased judgments.
VerificationIndicators	IIDVerificationIndicators	The IIDVerificationIndicators class contains the verification information that is returned for InstantID business searches.
VerifiedInput	IIDInput	The IIDInput class contains the input information and the verified information.
Watchlist	IIDWatchlist	This property is always null.

Related Information

[ResultError Class](#) on page 187

[IIDFEINMatch Class](#) on page 72

[IIDInput Class](#) on page 73

[IIDFiling Class](#) on page 72

[IIDReversePhone Class](#) on page 74

[IIDRiskIndicator Class](#) on page 74

[IIDVerificationIndicators Class](#) on page 75

[IIDWatchlist Class](#) on page 75

InstantIDIndividualResults Class

The InstantIDIndividualResults class contains the results for InstantID individual searches.

For possible results, see the LexisNexis® *InstantID® Analyzing Results* white paper at: https://bridgetinsight.custhelp.com/app/answers/detail/a_id/1442.

Class Properties

Name	Type	Description
AddressHistory	List<IIDAddressHistory>	The IIDAddressHistory class contains the other addresses associated with the input entity.
AlternateNames	List<IIDAlternate Name>	The IIDAlternateName class contains an alternative last name that is associated with the InstantID input entity.
Comprehensive VerificationIndex	Integer	This property contains the overall risk index (0, 10, 20, 30, 40, or 50).
Errors	List<ResultError>	The ResultError class returns any errors that were generated during a search.
Input	IIDInput	The IIDInput class contains the input information and the verified information.
NameAddressPhone	IIDRiskIndicator	The IIDRiskIndicator class contains the NAP results.
NameAddressSSN	IIDRiskIndicator	The IIDRiskIndicator class contains the NAS results.
PhoneOfNameAddress	String	This property contains the phone number that is associated with the input name and the input address.
PotentialFollowupActions	List<IIDRiskIndicator>	The IIDRiskIndicator class contains the risk indicators and the red flags.
RedFlags	List<IIDRedFlag>	The IIDRedFlag class contains the red flags.
ReversePhone	IIDReversePhone	The IIDReversePhone class contains the name and the address that are associated with the input phone number.
RiskIndicators	List<IIDRiskIndicator>	The IIDRiskIndicator class contains the risk indicators.
SSNInfo	IIDSSNInfo	The IIDSSNInfo class contains the SSN result information.
VerifiedInput	IIDInput	The IIDInput class contains the input information and the verified information.
Watchlist	IIDWatchlist	This property is always null.

Related Information

[IIDAddressHistory Class](#) on page 71

[IIDAlternateName Class](#) on page 71

[ResultError Class](#) on page 187

[IIDInput Class](#) on page 73

[IIDRiskIndicator Class](#) on page 74

[IIDSSNInfo Class](#) on page 74

[IIDRedFlag Class](#) on page 73

[DOB Match Levels](#) on page 79

[IIDReversePhone Class](#) on page 74

[IIDWatchlist Class](#) on page 75

DOB Match Levels

The DOB match level indicates the level of the match between the input DOB and the DOB found for the input entity.

Possible DOB Match Levels

Web Service Value	Description
0	No DOB found or no DOB submitted
1	Nothing matches
2	Only Day matches
3	Only Month matches
4	Only Day and Month match
5	Only Day and Year match
6	Only Year matches
7	Only Month and Year match
8	Month, Day, and Year match

List Screening Result Classes

The search response includes the classes that contain the list screening results.

Related Information

[Search Response Class Hierarchy](#) on page 56

OFACInstitutionInfo Class

The OFACInstitutionInfo class contains the information of the originating institution, the intermediary institution, or the beneficiary institution listed in the OFAC report.

Class Properties

Name	Type	Description
Address	String	This property contains the address.
Name	String	This property contains the name.
Type	String	This property contains the type.

OFACReportInfo Class

The OFACReportInfo class contains the OFAC report information.

An OFAC report can be generated in the XG user interface for each alert that contains potential list matches against the OFAC lists. For more information, see the *LexisNexis® Bridger Insight® XG Help*.

Class Properties

Name	Type	Description
ActionDate	String	This property contains the date that the action was taken.
AdditionalData	String	This property contains additional data.
AdditionalRelevantInfo	String	This property contains additional relevant information.
Amount	String	This property contains the amount of the transaction.
Institutions	List<OFACInstitutionInfo>	The OFACInstitutionInfo class contains the information of the originating institution, the intermediary institution, or the beneficiary institution listed in the OFAC report.
NameOfSigner	String	This property contains the name of the signer.
PreparerDate	String	This property contains the preparation date of the report.
Reason	String	This property contains the reason for the report.
ReportType	OFACReportType	This property contains the report type. Possible values: - Blocked - Rejected
TitleOfSigner	String	This property contains the title of the signer.
TransactionDate	String	This property contains the transaction date.

Related Information

[OFACInstitutionInfo Class](#) on page 79

WatchlistResults Class

The WatchlistResults class indicates the status of the list search and contains potential matches.

Class Properties

Name	Type	Description
Matches	List<WLMatch>	The WLMatch class contains the list match information.
Status	WLSearchStatusType	This property contains the status that indicates whether a list screening search was performed and whether results were returned. Possible values: - NotSearched - NoResults - Results

Related Information

[WLMatch Class](#) on page 84

WLAddressMatch Class

The WLAddressMatch class contains the address match information.

Class Properties

Name	Type	Description
InputValue	String	This property contains the input address.
ListValue	String	This property contains the address for the list entity.
Score	Integer	This property contains the address match score (0–100) that is used to help calculate the overall match confidence score.
Type	AddressType	This property contains the address type. Possible values: - None - Current - Mailing - Previous - Unknown None is not valid in a request. This property is required.

WLCitizenshipMatch Class

The WLCitizenshipMatch class contains the citizenship information for list entity matches.

Class Properties

Name	Type	Description
InputValue	String	This property contains the input citizenship value.
ListValue	String	This property contains the citizenship.
Score	Integer	This property contains the citizenship score (0–100) for the list entity.

WLCountryDetails Class

The WLCountryDetails class contains the best match information and details for a list country match.

Class Properties

Name	Type	Description
AKAs	List<String>	This property contains AKA values that are associated with the country match.
Cities	List<String>	This property contains the cities that are associated with the country match.
Codes	List<String>	This property contains the country codes that are associated with the country match.
Comments	String	This property contains additional uncategorized information about the country match. This information may include source contact information and web addresses.
Country	String	This property contains the country in the list record.
DateListed	String	This property contains the date that the country was added to the list.
ListReferenceNumber	String	This property contains the source list reference number that is used by LexisNexis.
Ports	List<String>	This property contains the ports that are associated with the country match.
ReasonListed	String	This property contains the reason the country was added to the list.
Terms	List<String>	This property contains the country terms that are associated with the country match.

WLDOBMatch Class

The WLDOBMatch class contains the DOB information for list entity matches.

Class Properties

Name	Type	Description
InputValue	String	This property contains the input DOB.

Class Properties (Continued)

Name	Type	Description
ListValue	String	This property contains the DOB for the list entity.
Score	Integer	This property contains the DOB match score (0–100) that is used to help calculate the overall match confidence score.

WLEntityDetails Class

The WLEntityDetails class contains the screening list information, including the list record that best matched your input entity.

Class Properties

Name	Type	Description
AdditionalInfo	List<EntityAdditionalInfo>	The EntityAdditionalInfo class contains the additional information that helps identify the list entity.
Addresses	List<EntityAddress>	The EntityAddress class contains the address information.
AKAs	List<EntityAKA>	The EntityAKA class contains the alternative name information.
Comments	String	This property contains uncategorized additional information about the list entity. This information may include source contact information, web addresses, or email addresses.
DateListed	String	This property contains the date that the entity was added to the list.
EntityType	ListEntityType	This property contains the entity type. Possible values: - None - Business - Individual - Unknown - Vessel None is not valid in a request.
Gender	String	This property contains the entity gender type. Possible values: - None - Female - Male - Unknown None is not valid in a request.
IDs	List<EntityID>	The EntityID class contains the entity ID information.

Class Properties (Continued)

Name	Type	Description
ListReferenceNumber	String	This property contains the source list reference number that is used by LexisNexis.
Name	EntityName	The EntityName class contains the name information.
Phones	List<EntityPhone>	The EntityPhone class contains the phone number information.
ReasonListed	String	This property contains the reason that the entity was added to the list.

Related Information

[EntityAdditionalInfo Class](#) on page 174

[EntityAddress Class](#) on page 174

[EntityAKA Class](#) on page 175

[EntityID Class](#) on page 176

[EntityName Class](#) on page 176

[EntityPhone Class](#) on page 177

WLIDMatch Class

The WLIDMatch class contains the ID information for list entity matches.

Class Properties

Name	Type	Description
InputValue	String	This property contains the input ID.
ListValue	String	This property contains the ID for the list entity.
Score	Integer	This property contains the ID match score (0–100) that is used to help calculate the overall match confidence score.
Type	IDType	This property contains the ID type. This property is required.

Related Information

[Possible ID Types](#) on page 180

WLMatch Class

The WLMatch class contains the list match information.

Class Properties

Name	Type	Description
AcceptListID	Long	This property contains a system-generated number that identifies the accept list the alert was added to.
AddedToAcceptList	Boolean	True indicates that the alert was added to an accept list.
Addresses	List<WLAddressMatch>	The WLAddressMatch class contains the address match information. This property is returned only for matches on list entity records.
AddressName	Boolean	This property contains the address name.
AutoFalsePositive	Boolean	True indicates that XG set the WorldCompliance match to the False Positive disposition option because another WorldCompliance match in the alert was set to True Match. XG was configured to handle multiple matches in this manner using the TrueMatchSetsOthersToFalsePositive property.
BestCountry	String	This property contains the country that most closely matched the input information.
BestCountryScore	Integer	This property contains the score of the country that most closely matched the input information.  If a valid score is not available, -1 is returned.
BestCountryType	CountryPhraseType	This property contains the type of the BestCountry property. Possible values: • None • AlternateName • City • Code • Port • Term None is not valid in a request.
BestDOBIsPartial	Boolean	True indicates that the best DOB match was to a partial DOB.
BestName	String	This property contains the entity name that most closely matched the input information.
BestNameScore	Integer	This property contains the score of the name that most closely matched the input information.  If a valid score is not available, -1 is returned.
Checksum	Integer	A system-generated number that is used by XG to determine if True Match records have changed.

Class Properties (Continued)

Name	Type	Description
Citizenships	List<WLCitizenshipMatch>	The WLCitizenshipMatch class contains the citizenship information for list entity matches.
Conflicts	WLMatchConflicts	The WLMatchConflicts class contains the match conflicts.
CountryDetails	WLCountryDetails	The WLCountryDetails class contains the best match information and details for a list country match. This property is returned only for matches on list country records.
DOBs	List<WLDOBMatch>	The WLDOBMatch class contains the DOB information for list entity matches. This property is returned only for matches on list entity records.
EntityDetails	WLEntityDetails	The WLEntityDetails class contains the screening list information, including the list record that best matched your input entity. This property is returned only for matches on list entity records.
EntityName	String	This property contains the main entity name from the list record that generated the potential match.
EntityScore	Integer	This property contains the match confidence score (0-100) that XG generated to indicate how closely the list record matched the input entity information.  <i>If a valid score is not available, -1 is returned.</i>
EntityUniqueID	String	This property contains a system-generated number that identifies the entity.
Error	ResultError	The ResultError class returns any errors that were generated during a search.
FalseMatchUpdate	FalseMatchUpdateType	This property contains the type of data element that generated a false match update. Possible values: • None • BestName • DOB • Gender • AKA • Ids • Address • Phone • Citizenship
FalsePositive	Boolean	True indicates that the match decision was set to False Positive.
File	WLMatchFile	The WLMatchFile class contains the information about the list that contained the potential match.

Class Properties (Continued)

Name	Type	Description
GatewayOFACScreeningIndicator	Boolean	True indicates that a third-party service found OFAC matches before the record was searched in XG.
ID	Long	This property contains a system-generated number that identifies the match.
IDs	List<WLIDMatch>	The WLIDMatch class contains the ID information for list entity matches. This property is returned only for matches on list entity records.
MatchReAlert	Boolean	True indicates that this match is the result of a match re-alert. A match re-alert occurs when the match information in the list has changed.
MatchXML	WLMatchSummary	The WLMatchSummary class contains the summary of a potential list match when data elements other than name were used to find the match. This property is returned only for alerts with potential list matches on data elements other than name.
MatchIMDSDetails	WLMatchFprDetails	The WLMatchFprDetails class contains the results from IMDS.
OFACInfo	OFACReportInfo	The OFACReportInfo class contains the OFAC report information.
Phones	List<WLPhoneMatch>	The WLPhoneMatch class contains the phone information for list entity matches. This property is returned only for matches on list entity records.
PreviousResultID	Long	This property contains the previous result ID.
ReasonListed	String	This property contains the reason that the entity was listed in the list.
ResultDate	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that XG generated the alert.
SecondaryOFACScreeningIndicator	Boolean	True indicates that a third-party service found OFAC matches before the record was searched in XG.
TrueMatch	Boolean	True indicates that the match decision was set to True Match.

Related Information

[WLAddressMatch Class](#) on page 81

[WLCitizenshipMatch Class](#) on page 81

[WLMatchConflicts Class](#) on page 88

[WLCountryDetails Class](#) on page 82

[WLDOBMatch Class](#) on page 82

[WLEntityDetails Class](#) on page 83

[ResultError Class](#) on page 187

[WLMatchFile Class](#) on page 88

[WLIDMatch Class](#) on page 84

[WLMatchSummary Class](#) on page 89

[OFACReportInfo Class](#) on page 80

[WLPhoneMatch Class](#) on page 90

WLMatchConflicts Class

The WLMatchConflicts class contains the match conflicts.

Class Properties

Name	Type	Description
AddressConflict	Boolean	True indicates that the input address and the address for the list entity did not match.
CitizenshipConflict	Boolean	True indicates that the input citizenship and the citizenship for the list entity did not match.
CountryConflict	Boolean	True indicates that the input country and the country for the list entity did not match.
DOBConflict	Boolean	True indicates that the input DOB and the DOB for the list entity did not match.
EntityTypeConflict	Boolean	True indicates that the input entity type and the entity type of the list entity did not match.
GenderConflict	Boolean	True indicates that the input gender and gender for the list entity did not match.
IDConflict	Boolean	True indicates that the input ID and the ID for the list entity did not match.
PhoneConflict	Boolean	True indicates that the input phone and the phone for the list entity did not match.

WLMatchFile Class

The WLMatchFile class contains the information about the list that contained the potential match.

Class Properties

Name	Type	Description
Build	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that LexisNexis built the list.
Custom	Boolean	True indicates that the list is a custom screening list.
ID	Unsigned Integer	This property contains a system-generated number that identifies the list.

Class Properties (Continued)

Name	Type	Description
Name	String	This property contains the list name.
Published	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that LexisNexis published the list.
Type	DataFileType	This property contains the data file type. Possible values: - UDF (XG file) - ADF (accept list) - BDF (entity list) - CDF (country list) - SDF (XG file)

WLMATCHFPRDETAILS Class

The WLMATCHFPRDETAILS class contains the results from IMDS.

Class Properties

Name	Type	Description
IsIMDSFalsePositive	Boolean	This property indicates whether the match was determined to be an automatic false positive.
Confidence	Integer	This property contains the false positive confidence score that was determined for the match.
ReasonCodes	String	This property contains the reason codes that were returned for the match.
RulesName	String	This property contains the rules for IMDS that were used in the search.
Version	String	This property contains the version of the rules for IMDS that were used in the search.

WLMATCHSUMMARY Class

The WLMATCHSUMMARY class contains the summary of a potential list match when data elements other than name were used to find the match.

Class Properties

Name	Type	Description
AddressMatches	List<WLMATCHSUMMARY Item>	The WLMATCHSUMMARYItem class contains the data element match summary of a potential list match on data elements other than the name.

Class Properties (Continued)

Name	Type	Description
CitizenshipMatches	List<WLMatchSummaryItem>	The WLMatchSummaryItem class contains the data element match summary of a potential list match on data elements other than the name.
DOBMatches	List<WLMatchSummaryItem>	The WLMatchSummaryItem class contains the data element match summary of a potential list match on data elements other than the name.
Gender	WLMatchSummaryItem	The WLMatchSummaryItem class contains the data element match summary of a potential list match on data elements other than the name.
IDMatches	List<WLMatchSummaryItem>	The WLMatchSummaryItem class contains the data element match summary of a potential list match on data elements other than the name.
PhoneMatches	List<WLMatchSummaryItem>	The WLMatchSummaryItem class contains the data element match summary of a potential list match on data elements other than the name.

Related Information

[WLMatchSummaryItem Class](#) on page 90

WLMatchSummaryItem Class

The WLMatchSummaryItem class contains the data element match summary of a potential list match on data elements other than the name.

Class Properties

Name	Type	Description
Input	String	This property contains the input information.
List	String	This property contains the list match information.
Score	Integer	This property contains the data element score that helps calculate the overall match confidence score.
Type	String	This property contains the specific data type. Examples of data type include current address, mailing address, or passport.

WLPhoneMatch Class

The WLPhoneMatch class contains the phone information for list entity matches.

Class Properties

Name	Type	Description
InputValue	String	This property contains the input phone number.
ListValue	String	This property contains the phone number for the list entity.
Score	Integer	This property contains the phone match score (0–100) that is used to help calculate the overall match confidence score.
Type	PhoneType	This property contains the phone type. Possible values: • Business • Cell • Fax • Home • Work • Unknown • None None is not valid in a request. This property is required.

Chapter 5

Results Interface

LexisNexis[®] Bridger Insight[®] XG (XG) web services provide a Results interface that lets you manage your alerts. You can also manage alerts through the XG application.

For more information, see the *LexisNexis[®] Bridger Insight[®] XG Help*.

Run Management

The Results interface lets you manage the runs that are stored in the XG database. A run contains the details about a search that was performed and the alerts that were generated.

The Results interface lets you perform the following actions on runs:

- View a run.
- Search for a run.
- Delete a run.

Related Information

[C# Code Examples](#) on page 192

DeleteRuns Method

The DeleteRuns method deletes the specified runs.

Syntax

```
List<long> DeleteRuns(ClientContext context, List<long> runIDs)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

runIDs

The parameter contains the system-generated numbers that identify the runs.

To obtain a run ID, you use the SearchRuns method to search for the run.

Return Values

If all the runs are deleted, the response returns `Null`.

If any runs were not deleted, the response returns a list of the run IDs that identify the runs that were not deleted.

Related Information

[ClientContext Class](#) on page 21

[SearchRuns Method](#) on page 94

[DeleteRuns Code Example](#) on page 209

GetRunInfo Method

The GetRunInfo method returns the information about the specified run.

Syntax

```
RunInfo GetRunInfo(ClientContext context, long runID)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

runID

The parameter contains the system-generated number that identifies the run.

To obtain a run ID, you use the SearchRuns method to search for the run.

Return Values

The RunInfo class contains the run information.

Related Information

[ClientContext Class](#) on page 21

[SearchRuns Method](#) on page 94

[RunInfo Class](#) on page 96

[GetRunInfo Code Example](#) on page 207

SearchRuns Method

The SearchRuns method returns the runs that meet the specified criteria.

Syntax

```
List<RunInfo> SearchRuns(ClientContext context, RunsCriteria criteria)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

criteria

The RunsCriteria class defines the criteria a run must meet to be returned in a query.

Return Values

The RunInfo class contains the run information.

Related Information

[ClientContext Class](#) on page 21

[RunsCriteria Class](#) on page 95

[RunInfo Class](#) on page 96

[SearchRuns Code Example](#) on page 208

Run Management Classes

The classes that are returned in the request or the response vary depending on the run management method. Some classes contain other classes as properties. The classes are described in alphabetical order. Classes that are used in multiple web services interfaces may be described in another section.

Run Management Class Hierarchy

A class hierarchy depicts the structure of the classes within the code.

The following class hierarchy depicts the structure of the run management classes:

"ClientContext Class" on page 21

"RunsCriteria Class" on page 95 > "Date Class" on page 95

"RunInfo Class" on page 96

Date Class

The Date class contains date information.

Class Properties

Name	Type	Description
Day	Integer	This property contains the day value in a structured date.
Month	Integer	This property contains the month value in a structured date.
Unparsed	String	This property contains the unstructured date value.
Year	Integer	This property contains the year value in a structured date.

RunsCriteria Class

The RunsCriteria class defines the criteria a run must meet to be returned in a query. Runs must meet the criteria from each specified property to be returned.

If multiple **SubmitType** options are specified, runs that meet any of the possible values provided in the request are returned.

Class Properties

Name	Type	Description
DateEnd	Date	This property contains the latest date that the run was performed. The Date class contains date information.
DateStart	Date	This property contains the earliest date that the run was performed. The Date class contains date information.
Type	EFTType	This property contains the payment transaction type. Possible values: - None - ACH - Fedwire - ISO20022 - SWIFT - Unstructured None is not valid in a request. XML in ISO20022 requests must be encoded before processing.

Class Properties (Continued)

Name	Type	Description
EntityType	ResultEntityType	This property contains the entity type. Possible values: • Business • Individual • Text • Unknown • Unstructured
FileName	String	This property contains the name of the batch file that was searched.
PredefinedSearch	String	This property contains the name of the predefined search.
ProcessingState	ProcessingStateType	This property contains the processing state. Possible values: • None • Canceled • Completed • Error • Incomplete • Interrupted • Processing
SubmitType	RunsCriteriaSubmitType	This property contains the run type. Possible values: • All • AutomaticBatch • Batch • Import • None • RealTime • WebServices You can provide more than one SubmitType property. Runs that meet any of the possible values provided in the request are returned.

Related Information

[Date Class](#) on page 95

RunInfo Class

The RunInfo class contains the run information.

Class Properties

Name	Type	Description
AlertCount	Integer	This property contains the number of alerts generated.
Division	String	This property contains the division that can access the search results.
DivisionID	Integer	This property contains a system-generated number that identifies the division.
DPPA	DPPAChoiceType	This property contains the DPPA (Driver's Privacy Protection Act) permissible use. Possible values: • NA • Choice0 • Choice1 • Choice2 • Choice3 • Choice4 • Choice6 The default value is NA. This value is not valid if a DPPA permissible use is required. This property is required for LexisNexis® FraudPoint® Score searches or LexisNexis® InstantID® searches on individual entities.
Type	EFTType	This property contains the payment transaction type. Possible values: • None • ACH • Fedwire • ISO20022 • SWIFT • Unstructured None is not valid in a request. XML in ISO20022 requests must be encoded before processing.
EntityType	ResultEntityType	This property contains the entity type. Possible values: • Business • Individual • Text • Unknown • Unstructured
ErrorCount	Integer	This property contains the number of errors that the search generated.
ErrorMessage	String	This property contains a system-generated error that prevented the run from being completed.

Class Properties (Continued)

Name	Type	Description
FileName	String	This property contains the batch file that was searched.
FileFormatName	String	A custom format that identified the fields in the batch file.
GLB	Integer	This property contains the GLBA (Gramm-Leach-Bliley Act) permissible use. Possible values: • NA • 0 • 1 • 3 • 5 • 6 • 7 • 12 The default value is NA. This value is not valid if a GLBA permissible use is required. This property is required for FraudPoint Score searches or InstantID searches.
LastUpdateTime	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that the run was updated.
NumRecordsProcessed	Integer	This property contains the number of records that were searched.
PredefinedSearchName	String	This property contains the name of the predefined search.
ProcessingState	ProcessingStateType	This property contains the processing state. Possible values: • None • Canceled • Completed • Error • Incomplete • Interrupted • Processing
ResultsFile1	String	This property contains the result file.
ResultsFile2	String	This property contains the result file.
RunID	Long	This property contains a system-generated number that identifies the run.
StartTime	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that the search was started.
SubmitType	String	This property contains the submit type.

Class Properties (Continued)

Name	Type	Description
TotalNumRecords	Integer	This property contains the number of records in the batch file.
TotalNumRows	Integer	This property contains the number of rows in the batch file.
UserName	String	This property contains the user name of the user who initiated the search.
WLMatchCount	Integer	This property contains the number of potential matches that were generated.

Result Management

The Results interface lets you manage the alerts that are stored in the XG database.

The Results interface lets you perform the following actions on alerts:

- View an alert that contains specific information.
- Delete an alert.
- Query for alerts that contain specific information.
- Set the record state for an alert.

Related Information

[C# Code Examples](#) on page 192

DeleteResultRecords Method

The DeleteResultRecords method deletes the specified alerts.

Syntax

```
List<long> DeleteResultRecords(ClientContext context, List<long> resultIDs)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

resultIDs

The parameter contains the system-generated numbers that identify the alerts to be deleted.

The maximum number of records that can be deleted in one request is 1,000.

To obtain a result ID, you use the `SearchResultRecords` method to search for the alert. The ID is also returned for the alert in a search response.

Return Values

If all the alerts were deleted, the response returns `Null`.

If any alerts were not deleted, the response returns a list of result IDs that identifies the alerts that were not deleted.

Related Information

[ClientContext Class](#) on page 21

[SearchResultRecords Method](#) on page 101

[ResultRecord Class](#) on page 62

[DeleteResultRecords Code Example](#) on page 209

GetResultRecords Method

The `GetResultRecords` method returns the specified alerts.

Syntax

```
List<ResultRecord> GetResultRecords(ClientContext context, List<long> resultIDs)
```

Parameters

The method contains the following parameters:

context

The `ClientContext` class contains the XG client information and the XG session information.

resultIDs

The parameter contains the system-generated numbers that identify the alerts to be returned.

The maximum number of alerts that can be returned in one request is 100.

Return Values

The response returns a list of the alerts that were returned.

Related Information

[ClientContext Class](#) on page 21

[GetResultRecords Code Example](#) on page 210

SearchResultRecords Method

The SearchResultRecords method returns the alerts that meet the specified criteria in the result query.

Syntax

```
SearchRecordResults SearchResultRecords(ClientContext context, ResultsCriteria  
criteria, long position, int numberToReturn)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

criteria

The ResultsCriteria class contains the criteria that alerts must meet to be returned in a result query.

position

The parameter contains the position of the first record ID to be returned.

numberToReturn

The parameter contains the number of record IDs, up to 100, to be returned.

Return Values

The SearchRecordResults class contains the record IDs of the alerts that met the criteria in a result query.

If no records were found or the specified position is beyond the end of the matching record, the response returns an empty list.

Related Information

[ClientContext Class](#) on page 21

[ResultsCriteria Class](#) on page 112

[SearchRecordResults Class](#) on page 117

[SearchResultRecords Code Example](#) on page 212

SetRecordState Method

The SetRecordState method changes the state of an alert.

Syntax

```
bool SetRecordState(ClientContext context, long resultID, RecordState state)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

resultID

The parameter contains the system-generated number that identifies the alert.

To obtain a result ID, you use the SearchResultRecords method to search for the alert. The ID is also returned for the alert in a search response.

state

The RecordState class contains the alert state information.

Return Values

The response always returns True.

If the method fails or there is an error, then the response also returns a ServiceFault class.

Related Information

[ClientContext Class](#) on page 21

[SearchResultRecords Method](#) on page 101

[ResultRecord Class](#) on page 62

[RecordState Class](#) on page 61

[ServiceFault Class](#) on page 181

[SetRecordState Code Example](#) on page 211

Result Management Classes

The classes that are returned in the request or the response vary depending on the result management method. Some classes contain other classes as properties. The classes are described in alphabetical order. Classes that are used in multiple web services interfaces may be described in another section.

Result Management Class Hierarchy

A class hierarchy depicts the structure of the classes within the code.

The following class hierarchy depicts the structure of the result management classes:

"ClientContext Class" on page 21

"RecordState Class" on page 61:

- "AuditItem Class" on page 60
- "WatchlistMatchState Class" on page 67

"ResultRecord Class" on page 62:

- "FraudPointResults Class" on page 69:
 - "FPInput Class" on page 68:
 - "FPAddress Class" on page 68
 - "InputName Class" on page 172
 - "FPRedFlag Class" on page 68 > "FPRiskIndicator Class" on page 69
 - "FPRiskIndicator Class" on page 69
 - "ResultError Class" on page 187
- "InstantIDBusinessResults Class" on page 76:
 - "IIDFEINMatch Class" on page 72 > "IIDAddress Class" on page 71
 - "IIDFiling Class" on page 72 > "IIDAddress Class" on page 71
 - "IIDInput Class" on page 73:
 - "IIDAddress Class" on page 71
 - "InputName Class" on page 172
 - "IIDReversePhone Class" on page 74:
 - "IIDAddress Class" on page 71
 - "InputName Class" on page 172
 - "IIDRiskIndicator Class" on page 74
 - "IIDVerificationIndicators Class" on page 75
 - "IIDWatchlist Class" on page 75 > "IIDAddress Class" on page 71
 - "ResultError Class" on page 187
- "InstantIDIndividualResults Class" on page 77:
 - "IIDAddressHistory Class" on page 71 > "IIDAddress Class" on page 71
 - "IIDAlternateName Class" on page 71
 - "IIDInput Class" on page 73:
 - "IIDAddress Class" on page 71
 - "InputName Class" on page 172
 - "IIDRedFlag Class" on page 73 > "IIDRiskIndicator Class" on page 74
 - "IIDReversePhone Class" on page 74:
 - "IIDAddress Class" on page 71
 - "InputName Class" on page 172
 - "IIDRiskIndicator Class" on page 74
 - "IIDSSNInfo Class" on page 74
 - "IIDWatchlist Class" on page 75 > "IIDAddress Class" on page 71

- "ResultError Class" on page 187

"ResultsCriteria Class" on page 112:

- "Date Class" on page 95
- "EntityCriteriaAdditionalInfo Class" on page 178
- "EntityCriteriaAddress Class" on page 178
- "EntityCriteriaID Class" on page 179
- "FraudPointResultCriteria Class" on page 106 > "FPResultCriteriaRedFlag Class" on page 105
- "InstantIDResultCriteria Class" on page 111:
 - "IIDResultCriteriaBusiness Class" on page 108
 - "IIDResultCriteriaIndividual Class" on page 109 > "IIDResultCriteriaRedFlag Class" on page 110
- "ResultCriteriaAccount Class" on page 114
- "ResultCriteriaEFT Class" on page 114
- "ResultCriteriaRecordState Class" on page 115 > "Date Class" on page 95
- "WatchlistResultCriteria Class" on page 117

"SearchRecordResults Class" on page 117

Date Class

The Date class contains date information.

Class Properties

Name	Type	Description
Day	Integer	This property contains the day value in a structured date.
Month	Integer	This property contains the month value in a structured date.
Unparsed	String	This property contains the unstructured date value.
Year	Integer	This property contains the year value in a structured date.

FPResultCriteriaRedFlag Class

The FPResultCriteriaRedFlag class contains red flag criteria that an alert must meet to be returned in a result query.

Class Properties

Name	Type	Description
PrimaryID	Integer	This property contains the primary ID.
SecondaryID	Integer	This property contains the secondary ID.

FraudPointResultCriteria Class

The FraudPointResultCriteria class contains the FraudPoint Score criteria that an alert must meet to be returned in a result query.

For possible risk indicators and scores, see the *LexisNexis® FraudPoint® Score Analyzing Results* white paper on the LexisNexis Technical Support website at: https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1441.

Class Properties

Name	Type	Description
ErrorText	String	This property contains the error text.
FriendlyFraudIndexStart	Integer	This property contains the start value of the index range to be queried. The index range is 1-9.
FriendlyFraudIndexEnd	Integer	This property contains the end value of the index range to be queried.
ManipulatedIdentityIndexStart	Integer	This property contains the start value of the index range to be queried. The index range is 1-9.
ManipulatedIdentityIndexEnd	Integer	This property contains the end value of the index range to be queried. The index range is 1-9.
ProviderErrorsOnly	Boolean	True includes only alerts in which the FraudPoint Score search generated an error. The default value is false.
RedFlags	List<FPResultCriteriaRedFlag>	The FPResultCriteriaRedFlag class contains red flag criteria that an alert must meet to be returned in a result query.
RiskIndicators	List<Integer>	This property contains an integer list of the risk indicators.
ScoreEnd	Integer	This property contains the highest score that the alert can have and still be returned.
ScoreStart	Integer	This property contains the lowest score that the alert can have and still be returned.
StolenIdentityIndexStart	Integer	This property contains the start value of the index range. The index range is 1-9.

Class Properties (Continued)

Name	Type	Description
StolenIdentityIndexEnd	Integer	This property contains the end value of the index range.
SuspiciousActivityIndexStart	Integer	This property contains the start value of the index range. The index range is 1-9.
SuspiciousActivityIndexEnd	Integer	This property contains the end value of the index range.
SyntheticIdentityIndexStart	Integer	This property contains the start value of the index range. The index range is 1-9.
SyntheticIdentityIndexEnd	Integer	This property contains the end value of the index range.
Version	FraudPointVersion	This property contains the LexisNexis® FraudPoint® version that was used to perform the search. Possible values: • None • V1_0 • V2_0
VulnerableVictimIndexStart	Integer	This property contains the start value of the index range. The index range is 1-9.
VulnerableVictimIndexEnd	Integer	This property contains the end value of the index range.

Related Information

[FPResultCriteriaRedFlag Class](#) on page 105

IIDIResultCriteriaIndividual Class

The IIDIResultCriteriaIndividual class contains the LexisNexis® InstantID® International criteria that alerts must meet to be returned in a result query.

For possible index values and risk indicators, see the *LexisNexis® InstantID® International Analyzing Results* white paper on the LexisNexis Technical Support website at: https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1443.

Class Properties

Name	Type	Description
IndexEnd	IndexLevelType	This property contains the highest index that an alert can have to be returned. Possible values: • None • Level0 • Level10 • Level20 • Level30 • Level40 • Level50 This property is required.
IndexStart	IndexLevelType	This property contains the lowest index that an alert can have to be returned. Possible values: • None • Level0 • Level10 • Level20 • Level30 • Level40 • Level50 This property is required.
RiskIndicators	List<Integer>	This property contains the risk code that is associated with the risk indicator.

IIDResultCriteriaBusiness Class

The IIDResultCriteriaBusiness class contains the InstantID criteria that the alerts for business entities must meet to be returned in a result query.

For possible risk indicators and indexes, see the *LexisNexis® InstantID® Analyzing Results* white paper on the LexisNexis Technical Support website at: https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1442.

Class Properties

Name	Type	Description
IndexEnd	IndexLevelType	This property contains the highest index that an alert can have to be returned. Possible values: • None • Level0 • Level10 • Level20 • Level30 • Level40 • Level50
IndexStart	IndexLevelType	This property contains the lowest index that an alert can have to be returned. Possible values: • None • Level0 • Level10 • Level20 • Level30 • Level40 • Level50 This property is required.
RiskIndicators	List<Integer>	This property contains the risk code that is associated with the risk indicator.

IIDResultCriteriaIndividual Class

The IIDResultCriteriaIndividual class contains the InstantID criteria that the alerts for individual entities must meet to be returned in an alert query.

For possible risk indicators and indexes, see the *LexisNexis® InstantID® Analyzing Results* white paper on the LexisNexis Technical Support website at: https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1442.

Class Properties

Name	Type	Description
IndexEnd	IndexLevelType	This property contains the highest index that an alert can have to be returned. Possible values: • None • Level0 • Level10 • Level20 • Level30 • Level40 • Level50
IndexStart	IndexLevelType	This property contains the lowest index that an alert can have to be returned. Possible values: • None • Level0 • Level10 • Level20 • Level30 • Level40 • Level50 This property is required.
NAPCriteria	Integer	This property contains the name-address-phone number verification index.
NASCriteria	Integer	This property contains the name-address-SSN number verification index.
RedFlags	List<IIDResultCriteriaRedFlag>	The IIDResultCriteriaRedFlag class contains the red flag criteria that alerts must meet to be returned in a query.
RiskIndicators	List<Integer>	This property contains the risk code that is associated with the risk indicator.

Related Information

[IIDResultCriteriaRedFlag Class](#) on page 110

IIDResultCriteriaRedFlag Class

The IIDResultCriteriaRedFlag class contains the red flag criteria that alerts must meet to be returned in a query.

For possible red flags, see the *LexisNexis® InstantID® Analyzing Results* white paper on the LexisNexis Technical Support website at: https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1442.

Class Properties

Name	Type	Description
PrimaryID	Integer	This property contains the primary ID.
SecondaryID	Integer	This property contains the secondary ID.

InstantIDIntlResultCriteria Class

The InstantIDIntlResultCriteria class contains the InstantID International criteria that alerts must meet to be returned in an alert query.

Class Properties

Name	Type	Description
ErrorText	String	This property contains the error text.
Individual	IIDResultCriteriaIndividual	The IIDResultCriteriaIndividual class contains the InstantID International criteria that alerts must meet to be returned in a result query.
ProviderErrorsOnly	Boolean	True includes only alerts in which the InstantID International search generated an error. The default value is false.

Related Information

[IIDResultCriteriaIndividual Class](#) on page 107

InstantIDResultCriteria Class

The InstantIDResultCriteria class contains the InstantID criteria that alerts must meet to be returned in an alert query.

Class Properties

Name	Type	Description
Business	IIDResultCriteriaBusiness	The IIDResultCriteriaBusiness class contains the InstantID criteria that the alerts for business entities must meet to be returned in a result query.
ErrorText	String	This property contains the error text.
Individual	IIDResultCriteriaIndividual	The IIDResultCriteriaIndividual class contains the InstantID criteria that the alerts for individual entities must meet to be returned in an alert query.
ProviderErrorsOnly	Boolean	True includes only alerts in which the InstantID search generated an error. The default value is false.

Related Information

[IIDResultCriteriaBusiness Class](#) on page 108

[IIDResultCriteriaIndividual Class](#) on page 109

ResultsCriteria Class

The ResultsCriteria class contains the criteria that alerts must meet to be returned in a result query.

Class Properties

Name	Type	Description
Account	ResultCriteriaAccount	The ResultCriteriaAccount class contains the account criteria that alerts must meet to be returned in a result query.
AdditionalInfo	EntityCriteriaAdditionalInfo	The EntityCriteriaAdditionalInfo class contains the additional information that the record must meet to be returned.
Address	EntityCriteriaAddress	The EntityCriteriaAddress class contains the address information and the phone information that the record must meet to be returned.
AlertDecisionName	String	This property contains the name of the alert decision that was applied to the alert.
Context	String	This property contains the text string in an unstructured input record that contained the match.
DateEnd	Date	This property contains the latest date that the alert was generated.
DateStart	Date	The earliest date that the alert was generated.
EFT	ResultCriteriaEFT	The ResultCriteriaEFT class contains the payment transaction criteria that alerts must meet to be returned in a result query.
EntityType	ResultEntityType	This property contains the entity type. Possible values: • Business • Individual • Text • Unknown • Unstructured
FirstName	String	This property contains the first name.
FraudPoint	FraudPointResultCriteria	The FraudPointResultCriteria class contains the FraudPoint Score criteria that an alert must meet to be returned in a result query.
FullName	String	This property contains the full name.
ID	EntityCriteriaID	The EntityCriteriaID class contains the ID information that the record must meet to be returned.

Class Properties (Continued)

Name	Type	Description
InstantID	InstantIDResultCriteria	The InstantIDResultCriteria class contains the InstantID criteria that alerts must meet to be returned in an alert query.
InstantIDIntl	InstantIDIntlResultCriteria	The InstantIDIntlResultCriteria class contains the InstantID International criteria that alerts must meet to be returned in an alert query.  <i>The InstantID International service is no longer available. Some XG 4 clients may still view service results in their database.</i>
LastName	String	This property contains the last name.
MaidenName	String	This property contains the maiden name.
MatchText	String	This property contains the match text.
MiddleName	String	This property contains the middle name.
RecordState	ResultCriteriaRecordState	The ResultCriteriaRecordState class contains the record state information that alerts must meet to be returned in a result query.
RunIDs	List<Long>	This property contains a system-generated number that identifies the run that generated the alert.
SecondSurname	String	The second surname.
SSNEIN	String	This property contains the SSN (Social Security number) or EIN (employer identification number).
TimeEnd	Time	This property contains the latest time the alert was generated.
TimeStart	Time	This property contains the earliest time the alert was generated.
Watchlist	WatchlistResultCriteria	The WatchlistResultCriteria class contains the list criteria that alerts must meet to be returned in a list query.

Related Information

[ResultCriteriaAccount Class](#) on page 114
[EntityCriteriaAdditionalInfo Class](#) on page 178
[EntityCriteriaAddress Class](#) on page 178
[Date Class](#) on page 95
[ResultCriteriaEFT Class](#) on page 114
[FraudPointResultCriteria Class](#) on page 106
[EntityCriteriaID Class](#) on page 179
[InstantIDResultCriteria Class](#) on page 111
[InstantIDIntlResultCriteria Class](#) on page 111
[ResultCriteriaRecordState Class](#) on page 115
[WatchlistResultCriteria Class](#) on page 117
[Time Class](#) on page 117

ResultCriteriaAccount Class

The ResultCriteriaAccount class contains the account criteria that alerts must meet to be returned in a result query.

Class Properties

Name	Type	Description
Amount	String	This property contains the account amount.
Date	String	This property contains the account date.
GroupID	String	This property contains the account group ID.
ID	String	This property contains the account ID.
MemberID	String	This property contains the account member ID.
OtherData	String	This property contains the other account data that the alerts must contain to be returned.
ProviderID	String	This property contains the account provider ID.
Type	String	This property contains the account type.

ResultCriteriaEFT Class

The ResultCriteriaEFT class contains the payment transaction criteria that alerts must meet to be returned in a result query.

Class Properties

Name	Type	Description
DollarAmount	String	The dollar amount.
DollarAmount Comparison	ComparisonType	The dollar amount comparison type. Possible values: - None - Is - IsGreaterThan - IsGreaterThanEqual - IsLessThan - IsLessThanOrEqual - IsNull
DUNS	String	The DUNS (Data Universal Numbering System) number.
ExternalOFACIndicator	String	The external OFAC (Office of Foreign Assets Control) indicator.

Class Properties (Continued)

Name	Type	Description
TransactionID	String	The transaction ID.
Type	EFTType	This property contains the payment transaction type. Possible values: • None • ACH • Fedwire • ISO20022 • SWIFT • Unstructured None is not valid in a request. XML in ISO20022 requests must be encoded before processing.

ResultCriteriaRecordState Class

The ResultCriteriaRecordState class contains the record state information that alerts must meet to be returned in a result query.

Class Properties

Name	Type	Description
AlertAge	AlertAgeType	This property contains the maximum alert age in days. Possible values: • None • Days10 • Days20 • Days30 • Days60 • Days90
AlertState	AlertStateType	This property contains the alert state. Possible values: • None • Closed • Open • Unassigned None is not valid in a request. Unassigned is only returned for alerts that did not contain results. For example, this value is set if the search generated an error but not an alert.
AnyErrors	Boolean	True returns alerts that contain errors of any type.

Class Properties (Continued)

Name	Type	Description
AssignedTo	String	This property contains the user or the role that is assigned to the alert.
AssignmentType	AssignmentType	This property contains the assignment type. Possible values: • Role • Unknown • User
CurrentStatus	String	This property contains the status currently set for the alert.
Divisions	List<String>	This property contains the divisions that can access the alerts.
DateStateAssignedEnd	Date	This property contains the latest date that the alert entered the current state.
DateStateAssignedStart	Date	This property contains the earliest date that the alert entered the current state.
DateStatusAssignedEnd	Date	This property contains the latest date that the current status was set.
DateStatusAssignedStart	Date	This property contains the earliest date that the current status was set.
GeneralSearchErrorsOnly	Boolean	True returns alerts that contain only general search errors. The default value is false.
PredefinedSearch	String	This property contains the predefined search that defined the settings used in the search.
RecordChangedBy	String	This property contains the user name of the user who changed the status.
RecordHasAttachments	Boolean	True returns alerts that have an attachment. The default value is false.
ResultsWithoutDivision	Boolean	True returns alerts that are not restricted to a division. The default value is false.
ResultsWithoutStatus	Boolean	True returns alerts with no status. The default value is false.
StatusHistory	String	This property contains the status that was previously set for the alert.

Related Information

[Date Class](#) on page 95

SearchRecordResults Class

The SearchRecordResults class contains the record IDs of the alerts that met the criteria in a result query.

Class Properties

Name	Type	Description
RecordIDs	List<Long>	This property contains a list of record IDs that identify the records that were returned.
TotalNumRecords	Long	This property contains the total number of records that were returned.

Time Class

The Time class contains time information.

Class Properties

Name	Type	Description
Hour	Integer	This property contains the hour in a structured time.
Milliseconds	Integer	This property contains the milliseconds in a structured time.
Minute	Integer	This property contains the minute in a structured time.
Second	Integer	This property contains the seconds in a structured time.
Unparsed	String	This property contains the unstructured time value.

WatchlistResultCriteria Class

The WatchlistResultCriteria class contains the list criteria that alerts must meet to be returned in a list query.

Class Properties

Name	Type	Description
ErrorText	String	This property contains the error text.
ExcludeIfFalseMatchUpdatesOnly	Boolean	True sets XG to not return alerts that contain only false match updates. The default value is false.
ExcludeIfTrueMatchUpdatesOnly	Boolean	True sets XG to not return alerts that contain only true match updates. The default value is false.

Class Properties (Continued)

Name	Type	Description
FalseMatchUpdatesOnly	Boolean	True sets XG to return alerts that contain only false match updates. The default value is false.
MatchDispositionFalsePositive	Boolean	True sets XG to return alerts that contain a potential screening list match that is set as a false positive. The default value is false.
MatchDispositionTrueMatch	Boolean	True sets XG to return alerts that contain a potential screening list match that is set as a true match. The default value is false.
ReasonListed	String	This property contains the reason the entity was listed on the screening list.
ScoreEnd	Integer	This property contains the highest score that a potential list match can have for the alert to be returned.
ScoreStart	Integer	This property contains the lowest score that a potential list match can have for the alert to be returned.
SearchOnErrors	Boolean	True returns alerts with errors. The default value is false.
SourceName	String	This property contains the name of the list that contained the potential match.
TrueMatchUpdatesOnly	Boolean	True returns only the alerts with true match updates. The default value is false.

Attachment Management

The Results interface lets you manage the files that are attached to alerts.

The Results interface lets you perform the following actions on attachments:

- Add a file to an alert.
- View a file.
- Delete a file.

Related Information

[C# Code Examples](#) on page 192

AddAttachment Method

The AddAttachment method adds an attachment to a specified alert.

Syntax

```
long AddAttachment(ClientContext context, Attachment attachment)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

attachment

The Attachment class contains attachment information.

Return Values

The response returns the attachment ID number.

Related Information

[ClientContext Class](#) on page 21

[Attachment Class](#) on page 121

[GetResultRecords Method](#) on page 100

[AddAttachment Code Example](#) on page 213

DeleteAttachment Method

The DeleteAttachment method deletes a specified attachment from a specified alert.

Syntax

```
bool DeleteAttachment(ClientContext context, long attachmentID, long resultID)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

attachmentID

The parameter contains the system-generated number that identifies the file to be returned.

resultID

The parameter contains the system-generated number that identifies the alert that the file is attached to.

Return Values

The response always returns True.

If the method fails or there is an error, then the response also returns a ServiceFault class.

Related Information

[ClientContext Class](#) on page 21

[ServiceFault Class](#) on page 181

[DeleteAttachment Code Example](#) on page 214

GetAttachment Method

The GetAttachment method returns the information on the specified attachment.

Syntax

```
Attachment GetAttachment(ClientContext context, long attachmentID, long resultID)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

attachmentID

The parameter contains the system-generated number that identifies the file to be returned.

resultID

The parameter contains the system-generated number that identifies the alert that the file is attached to.

Return Values

The Attachment class contains attachment information.

Related Information

[ClientContext Class](#) on page 21

[Attachment Class](#) on page 121

[GetAttachment Code Sample](#) on page 215

Attachment Management Class

The classes that are provided in the request or returned in the response vary depending on the attachment management method that is used.

Attachment Class

The Attachment class contains attachment information.

Class Properties

Name	Type	Description
Data	Byte Array	This property contains the attachment data.
DateAdded	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that the attachment was added.
FileType	AttachmentFileType	This property contains the file type. Possible values: • None • GIF • JPE • TIF • BMP • HTM • JFIF • TIFF • CSV • HTML • MHT • TXT • DOC • JPEG • PDF • XLS • DOCX • JPG • PNG • XLSX  Although the property is not required, the FileType property is recommended.
ID	Long	This property contains a system-generated number that identifies the attachment
Name	String	This property contains the attachment name. This property is required.
Notes	String	This property contains user-defined information. This property is required when you add an attachment, if a note requirement was configured in the XG application.
ResultID	Long	This property contains a system-generated number that identifies the alert the file is attached to.
RunID	Long	This property contains a system-generated number that identifies the run that generated the alert the file is attached to.

Chapter 6

List Maintenance Interface

LexisNexis[®] Bridger Insight[®] XG (XG) web services provide a List Maintenance interface that you can use to manage accept lists and custom screening lists.

Accept lists use information that you added from previous XG searches to filter out list screening matches in new searches that you have already processed. Your organization has likely already determined that those matches were false positive or has determined that updates to the entity records in the match need to be monitored. For more information about how accept lists work, see the *Accept Lists and White List* white paper on the LexisNexis[®] Technical Support website at https://bridgerinsight.custhelp.com/app/answers/detail/a_id/1179.

Custom screening lists are used to flag countries or entities in list searches for purposes that are unique to your organization.

List Management

The List Maintenance interface lets you manage accept lists or custom screening lists.

The List Maintenance interface lets you perform the following actions on the lists:

- Add a list.
- Update a list.
- Delete a list.
- Search for lists that contain specific information.
- View a list.

XG 5 clients can also index custom screening lists.



Lists that are set to automatically index do not index when changes are made through XG web services. Before changes are reflected in XG, the list must be manually indexed in the XG application or using the `IndexList` method.

Related Information

[IndexList Method](#) on page 125

[C# Code Examples](#) on page 192

AddList Method

The AddList method creates a new accept list or custom screening list.

Records must be added to a custom screening list before the list can be included in a search. The XG 5 custom screening lists must be indexed before they can be included in a search.



XG Enterprise shared lists that need to be available to other product client instances must be created in the XG application.

Syntax

```
long AddList(ClientContext context, ListInfo list)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

list

The ListInfo class contains the list information when you add a list, update a list, view a list, or search for a list.

Return Values

The response returns the system primary key that identifies the list.

If the method fails or there is an error, the response returns a ServiceFault class.

Related Information

[ClientContext Class](#) on page 21

[ListInfo Class](#) on page 128

[ServiceFault Class](#) on page 181

[AddList Code Example](#) on page 200

DeleteList Method

The DeleteList method deletes the specified list.

You cannot delete the default accept list.

Before you delete a list, remove the list from the predefined searches in the XG application. You should also remove the list from the alert decisions. You may want to export the list before you delete the list to retain a copy for your audit records.



The changes that are made to a custom screening list are not available until the list is indexed.

Syntax

```
bool DeleteList(ClientContext context, long listID)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

listID

The parameter contains the system-generated primary key that identifies the list.

To obtain the list ID, you use the SearchLists method to search for the list.

Return Values

The response always returns True.

If the method fails or there is an error, then the response also returns a ServiceFault class.

Related Information

[ClientContext Class](#) on page 21

[SearchLists Method](#) on page 126

[ServiceFault Class](#) on page 181

[DeleteList Code Example](#) on page 201

GetList Method

The GetList method returns the details of the specified list.

Syntax

```
ListInfo GetList(ClientContext context, long listID)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

listID

The parameter contains the system-generated primary key that identifies the list.

To obtain the list ID, you use the SearchLists method to search for the list.

Return Values

The ListInfo class contains the list information when you add a list, update a list, view a list, or search for a list.

Related Information

[ClientContext Class](#) on page 21

[SearchLists Method](#) on page 126

[ListInfo Class](#) on page 128

[GetList Code Example](#) on page 202

IndexList Method

The IndexList method manually indexes the specified custom screening list.

Syntax

```
bool IndexList(ClientContext context, long listID)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

listID

The parameter contains the system-generated primary key that identifies the list.

To obtain the list ID, you use the SearchLists method to search for the list.

Return Values

The response always returns True.

If the method fails or there is an error, then the response also returns a ServiceFault class.

Related Information

[ClientContext Class](#) on page 21

[SearchLists Method](#) on page 126

[ServiceFault Class](#) on page 181

[IndexList Code Example](#) on page 201

SearchLists Method

The SearchLists method returns the lists that meet the specified criteria.

Syntax

```
List<ListInfo> SearchLists(ClientContext context, ListCriteria criteria)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

criteria

The ListCriteria class contains the criteria that a list must meet to be returned in a query.

Return Values

The ListInfo class contains the list information when you add a list, update a list, view a list, or search for a list.

Related Information

[ClientContext Class](#) on page 21

[ListCriteria Class](#) on page 127

[ListInfo Class](#) on page 128

[SearchLists Code Example](#) on page 199

UpdateList Method

The UpdateList method replaces the entire list entry in the XG database for the specified list.



The changes that are made to a custom screening list are not available until the list is indexed.

Syntax

```
bool UpdateList(ClientContext context, ListInfo list)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

list

The ListInfo class contains the list information when you add a list, update a list, view a list, or search for a list.



The properties that are already identified within the list are required when the product updates the list. Any existing properties that are not included in the request are deleted.

Return Values

The response always returns True.

If the method fails or there is an error, then the response also returns a ServiceFault class.

Related Information

[ClientContext Class](#) on page 21

[ListInfo Class](#) on page 128

[ServiceFault Class](#) on page 181

[UpdateList Code Example](#) on page 198

List Management Classes

The classes that are provided in the request or returned in the response vary depending on the list management method that is used. The classes are described in alphabetical order.

ListCriteria Class

The ListCriteria class contains the criteria that a list must meet to be returned in a query.



If multiple properties are included, the list must meet all criteria to be returned.

Class Properties

Name	Type	Description
Divisions	List<String>	This property contains a list of the divisions that can access the list.
ID	GUID	This property contains a system-generated primary key that identifies the list.

Class Properties (Continued)

Name	Type	Description
Name	String	This property contains the list name.
Type	ListType	This property contains the list type. Possible values: • None • Accept (List that contains accept list records) • Country (Custom screening list that contains country records) • Entity (Custom screening list that contains entity records) None is not valid in a request.

ListInfo Class

The ListInfo class contains the list information when you add a list, update a list, view a list, or search for a list.



When you update a list, the properties that are not included in the request are deleted.

Class Properties

Name	Type	Description
AccountNumbersUnique IDs	Boolean	True sets the product to treat account numbers as unique IDs. You use this property only for accept lists. The default value is false.
AutoIndex	Boolean	True sets the product to automatically index the list when changes are made in the XG application. The default value is false. <i>Lists that are set to automatically index do not index when changes are made through XG web services. The list must be manually indexed in the XG application or using the IndexList method before changes are reflected in the system.</i>
Comments	String	This property contains user-defined comments.
Description	String	This property contains the list description is required when you add a list.
Divisions	List<String>	This property contains a list of the divisions that can access the list. Only users in the specified divisions can make changes to the list. This restriction does not prevent other users from using the list to search.

Class Properties (Continued)

Name	Type	Description
Encrypt	Boolean	True sets the product to encrypt the list for security. The default value is false.
EntityUniqueIDType	IDType	This property contains the type of the unique ID. The product uses the unique ID to differentiate each record from other records with similar entity information.  Every record in the list must contain a unique ID with the selected ID type.
ID	Long	This property contains a system primary key you use to identify the list that you want to modify. If this property is included when you add a list, 0 is required. This property cannot be changed. This property is required.
Name	String	This property contains the list name is required when you add a list. This property is required.
Publication	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that LexisNexis published the list.
Type	ListType	This property contains the list type. Possible values: • None • Accept (List that contains accept list records) • Country (Custom screening list that contains country records) • Entity (Custom screening list that contains entity records) None is not valid in a request. This property is required.
UniqueID	GUID	This property contains a system-generated number that the product uses instead of the list name to identify the list. This property is required.
UserSelectedStatus Required	Boolean	True sets the product to require that a user set a status when an alert is added to the list. You use the property only for accept lists. The default value is false.

Related Information

[IndexList Method](#) on page 125

[Possible ID Types](#) on page 180

List Record Management

The List Maintenance interface lets you manage the records in accept lists or custom screening lists.

The List Maintenance interface lets you perform the following actions on list records:

- Add an alert to a list
- Update a list record.
- Delete a list record.
- View a list record.
- Search for list records that contain specific information.

You can also add records that you created to custom screening lists.



Lists that are set to automatically index do not index when changes are made through XG web services. Before changes are reflected in XG, the list must be manually indexed in the XG application or using the `IndexList` method.

Related Information

[IndexList Method](#) on page 125

[C# Code Examples](#) on page 192

AddRecords Method

The AddRecords method adds records that were not generated by a search to a specified custom screening list.

This method cannot be used to add records to accept lists.



The changes that are made to a custom screening list are not available until the list is indexed.

Syntax

```
List<AddRecordResult> AddRecords(ClientContext context, long listID,  
List<ListRecord> records)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

listID

The parameter contains the system-generated primary key that identifies the list.

To obtain the list ID, you use the SearchLists method to search for the list.

records

The ListRecord class contains the record information when you add a record, update a record, or view record information.

The request must contain all entity records or all country records. The type of record that is required depends on the type of custom screening list that you want to add the records to.

Return Values

The AddRecordResult class contains the information about the record that was added to the list.

Related Information

[ClientContext Class](#) on page 21

[SearchLists Method](#) on page 126

[ListRecord Class](#) on page 144

[AddRecordResult Class](#) on page 138

[AddRecord Code Example](#) on page 192

AddResultRecord Method

The AddResultRecord method adds a specified alert to an accept list or to an entity custom screening list. You cannot add an alert to a country custom screening list.

When you add an alert to an accept list, XG automatically changes the record as follows:

- The alert status is set to `FalsePositive`.
- The alert assignment is set to `Unassigned`.
- The alert state is set to `Closed`.



The changes that are made to a custom screening list are not available until the list is indexed.

Syntax

```
long AddResultRecord(ClientContext context, long listID, ResultInfo result)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

listID

The parameter contains the system-generated primary key that identifies the list.

To obtain the list ID, you use the SearchLists method to search for the list.

result

The ResultInfo class contains the alert information that XG used to create the list record.

Return Values

The response contains the system-generated primary key that identifies the list record.

Related Information

[ClientContext Class](#) on page 21

[SearchLists Method](#) on page 126

[ResultInfo Class](#) on page 146

[AddResultRecord Code Example](#) on page 193

DeleteRecords Method

The DeleteRecords method deletes specified records in a list. You may want to export the list before you delete the records to retain a copy for your audit records.



The changes that are made to a custom screening list are not available until the list is indexed.

Syntax

```
List<long> DeleteRecords(ClientContext context, long listID, List<long> recordIDs)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

listID

The parameter contains the system-generated primary key that identifies the list.

To obtain the list ID, you use the SearchLists method to search for the list.

recordIDs

A list of system-generated numbers that identify the records to be deleted.

The maximum number of records that can be deleted in one request is 100.

To obtain a record ID, you use the SearchListRecords method to search for the record.

Return Values

If all the records were successfully deleted, the response returns `Null`.

If any of the selected records were not deleted, the response returns a list of record IDs for the records that were not deleted.

Related Information

[ClientContext Class](#) on page 21

[SearchLists Method](#) on page 126

[SearchListRecords Method](#) on page 134

[DeleteRecords Code Example](#) on page 194

GetRecords Method

The GetRecords method returns the records from a specified accept list or custom screening list.

Syntax

```
List<ListRecord> GetRecords(ClientContext context, long listID, List<long> recordIDs)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

listID

The parameter contains the system-generated primary key that identifies the list.

To obtain the list ID, you use the SearchLists method to search for the list.

recordIDs

A list of system-generated numbers that identify the records to be returned.

The maximum number of records that can be deleted in one request is 100.

To obtain a record ID, you use the SearchListRecords method to search for the record.

Return Values

The ListRecord class contains the record information when you add a record, update a record, or view record information.

Related Information

[ClientContext Class](#) on page 21

[SearchLists Method](#) on page 126

[SearchListRecords Method](#) on page 134

[ListRecord Class](#) on page 144

[GetRecords Code Example](#) on page 196

SearchListRecords Method

The SearchListRecords method returns records in a specified list that meet the specified criteria.

Syntax

```
SearchRecordResults SearchListRecords(ClientContext context, long listID,  
ListRecordCriteria criteria, long position, int numberToReturn)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

listID

The parameter contains the system-generated primary key that identifies the list.

To obtain the list ID, you use the SearchLists method to search for the list.

criteria

The ListRecord class contains the record information when you add a record, update a record, or view record information.

position

The position, in the list, of the record ID of the first record to be returned.

numberToReturn

The number of record IDs, up to 100, to be returned.

Return Values

The SearchRecordResults class contains the record IDs of the alerts that met the criteria in a result query.

Related Information

[ClientContext Class](#) on page 21
[SearchLists Method](#) on page 126
[ListRecord Class](#) on page 144
[SearchRecordResults Class](#) on page 117
[SearchListRecords Code Example](#) on page 197

UpdateRecord Method

The UpdateRecord method replaces the entire record in the specified accept list record or custom screening list.



The changes that are made to a custom screening list are not available until the list is indexed.

Syntax

```
bool UpdateRecord(ClientContext context, ListRecord record)
```

Parameters

The method contains the following parameters:

context

The ClientContext class contains the XG client information and the XG session information.

record

The ListRecord class contains the record information when you add a record, update a record, or view record information.

Return Values

The response always returns True.

If the method fails or there is an error, then the response also returns a ServiceFault class.

Related Information

[SearchLists Method](#) on page 126
[ClientContext Class](#) on page 21
[ListRecord Class](#) on page 144
[ServiceFault Class](#) on page 181
[UpdateRecord Code Example](#) on page 195

List Record Management Classes

The classes that are provided in the request or returned in the response vary depending on the list record management method that is used. Some classes contain other classes as properties.

The classes are described in alphabetical order. Classes that are used in multiple web service interfaces may be described in another section.

Related Information

[IndexList Method](#) on page 125

[C# Code Examples](#) on page 192

List Record Management Class Hierarchy

A class hierarchy depicts the structure of the classes within the code.

The following class hierarchy depicts the structure of the list record management classes:

"AddRecordResult Class" on page 138

"ListRecordCriteria Class" on page 145:

- "CountryCriteria Class" on page 139
- "Date Class" on page 95
- "EntityCriteria Class" on page 141

"ListRecord Class" on page 144:

- "CountryInfo Class" on page 140:
 - "CountryAKA Class" on page 139
 - "CountryLocation Class" on page 140
- "EntityInfo Class" on page 142:
 - "EntityAdditionalInfo Class" on page 174
 - "EntityAddress Class" on page 174
 - "EntityAKA Class" on page 175 > "EntityName Class" on page 176
 - "EntityID Class" on page 176 > "Date Class" on page 95
 - "EntityMatchInfo Class" on page 143
 - "EntityName Class" on page 176
 - "EntityPhone Class" on page 177

"ResultInfo Class" on page 146 > "WatchlistMatchState Class" on page 67 and "ResultRecord Class" on page 62

The ResultRecord class contains the following classes as properties. Classes that contain information that is not added to a list record are not shown in this hierarchy. For example, classes that contain record state information and due diligence results are not shown.

- "ResultRecordDetails Class" on page 63:
 - "InputAdditionalInfo Class" on page 169 > "Date Class" on page 95
 - "InputAddress Class" on page 169
 - "InputID Class" on page 170 > "Date Class" on page 95
 - "InputName Class" on page 172
 - "ParsedACH Class" on page 148:
 - "ParsedACHField Class" on page 150
 - "ParsedACHNote Class" on page 150
 - "ParsedACHParty Class" on page 151:
 - "ParsedACHAdditionalInfo Class" on page 149
 - "ParsedACHAddress Class" on page 149
 - "ParsedACHID Class" on page 150
 - "ParsedFedwire Class" on page 154:
 - "ParsedFedwireField Class" on page 156 > "ParsedFedwireAmount Class" on page 155
 - "ParsedFedwireNote Class" on page 156
 - "ParsedFedwireParty Class" on page 157:
 - "ParsedFedwireAdditionalInfo Class" on page 154
 - "ParsedFedwireAddress Class" on page 154
 - "ParsedFedwireNote Class" on page 156
 - "ParsedFedwireID Class" on page 156
 - "ParsedFedwirePhone Class" on page 158
 - "ParsedISO20022 Class" on page 158:
 - "ParsedISO20022Field Class" on page 160
 - "ParsedISO20022Note Class" on page 161
 - "ParsedISO20022Party Class" on page 161:
 - "ParsedISO20022AdditionalInfo Class" on page 159
 - "ParsedISO20022Address Class" on page 159
 - "ParsedISO20022ID Class" on page 160
 - "ParsedISO20022Note Class" on page 161
 - "ParsedISO20022Phone Class" on page 162
 - "ParsedSWIFT Class" on page 163:
 - "ParsedSWIFTField Class" on page 165:
 - "ParsedSWIFTAmount Class" on page 165
 - "ParsedSWIFTOriginalMessage Class" on page 166
 - "ParsedSWIFTNote Class" on page 166

- "ParsedSWIFTParty Class" on page 167:
 - "ParsedSWIFTAdditionalInfo Class" on page 164
 - "ParsedSWIFTAddress Class" on page 164
 - "ParsedSWIFTID Class" on page 166
- "InputPhone Class" on page 172
- "RecordState Class" on page 61:
 - "AuditItem Class" on page 60
 - "WatchlistMatchState Class" on page 67
- "ResultError Class" on page 187
- "WatchlistResults Class" on page 80:
 - "OFACReportInfo Class" on page 80 > "OFACInstitutionInfo Class" on page 79
 - "ResultError Class" on page 187
 - "WLAddressMatch Class" on page 81
 - "WLCitizenshipMatch Class" on page 81
 - "WLCountryDetails Class" on page 82
 - "WLDOBMatch Class" on page 82
 - "WLEntityDetails Class" on page 83:
 - "EntityAdditionalInfo Class" on page 174
 - "EntityAddress Class" on page 174
 - "EntityAKA Class" on page 175
 - "EntityID Class" on page 176 > "Date Class" on page 95
 - "EntityName Class" on page 176
 - "EntityPhone Class" on page 177
 - "WLMatch Class" on page 84:
 - "WLIDMatch Class" on page 84
 - "WLMatchConflicts Class" on page 88
 - "WLMatchFile Class" on page 88
 - "WLMatchFprDetails Class" on page 89
 - "WLMatchSummary Class" on page 89 > "WLMatchSummaryItem Class" on page 90
 - "WLPhoneMatch Class" on page 90

"SearchRecordResults Class" on page 117

AddRecordResult Class

The AddRecordResult class contains the information about the record that was added to the list.

Class Properties

Name	Type	Description
ErrorMessage	String	This property contains the error message.
ID	Long	This property contains a system-generated number that identifies the record.

CountryAKA Class

The CountryAKA class contains the information for a country record.

Class Properties

Name	Type	Description
AKA	String	This property contains the AKA (also known as) value.
ID	Long	This property contains a system-generated number that identifies the entry in the record.
Type	CountryAKAType	This property contains the country AKA type. Possible values: • None • AKA • Code • Term None is not valid in a request. An example of a term is Iranian.

CountryCriteria Class

The CountryCriteria class contains the criteria a country record must meet to be returned in a list query.

Class Properties

Name	Type	Description
AlternateName	String	This property contains the country alternate name.
City	String	This property contains the city.
Code	String	This property contains the country code.
Name	String	This property contains the country name.

CountryInfo Class

The CountryInfo class contains the country information when you add a record, update a record, or view record information.

Class Properties

Name	Type	Description
AKAs	List<CountryAKA>	The CountryAKA class contains the information for a country record.
Locations	List<CountryLocation>	The CountryLocation class contains the location information for a country record.
Name	String	This property contains the country name.

Related Information

[CountryAKA Class](#) on page 139

[CountryLocation Class](#) on page 140

CountryLocation Class

The CountryLocation class contains the location information for a country record.

Class Properties

Name	Type	Description
ID	Long	This property contains a system-generated number that identifies the entry in the record.
Name	String	This property contains the country name.
Type	CountryLocationType	This property contains the country location type. Possible values: - None - City - Port None is not valid in a request.

Date Class

The Date class contains date information.

Class Properties

Name	Type	Description
Day	Integer	This property contains the day value in a structured date.
Month	Integer	This property contains the month value in a structured date.
Unparsed	String	This property contains the unstructured date value.
Year	Integer	This property contains the year value in a structured date.

EntityCriteria Class

The EntityCriteria class contains the criteria an entity record must meet to be returned in a list query.

Class Properties

Name	Type	Description
AdditionalInfo	EntityCriteriaAdditionalInfo	The EntityCriteriaAdditionalInfo class contains the additional information that the record must meet to be returned.
Address	EntityCriteriaAddress	The EntityCriteriaAddress class contains the address information and the phone information that the record must meet to be returned.
EntityAddedBy	String	This property contains the user name of the user who added the record to the list.
EntityType	ListEntityType	This property contains the entity type. Possible values: - None - Business - Individual - Unknown - Vessel None is not valid in a request.
ExcludeIfTrueMatchOnly	Boolean	True sets XG to not return records that contain only true match updates. You use this property to query only accept lists.
FirstName	String	This property contains the entity's first name.
FullName	String	This property contains the entity's full name.
ID	EntityCriteriaID	The EntityCriteriaID class contains the ID information that the record must meet to be returned.
LastName	String	This property contains the entity's last name.
MiddleName	String	This property contains the entity's middle name.

Class Properties (Continued)

Name	Type	Description
SSNEIN	String	This property contains the entity's SSN (Social Security number) or EIN (employer identification number).
TrueMatchesOnly	Boolean	True sets XG to return only records that contain only true match updates. You use this property to query accept lists only.

Related Information

[EntityCriteriaAdditionalInfo Class](#) on page 178

[EntityCriteriaAddress Class](#) on page 178

[EntityCriteriaID Class](#) on page 179

EntityInfo Class

The EntityInfo class contains the entity information when you add a record, update a record, or view record information.

Class Properties

Name	Type	Description
AdditionalInfo	List<EntityAdditionalInfo>	The EntityAdditionalInfo class contains the additional information that helps identify the list entity.
Addresses	List<EntityAddress>	The EntityAddress class contains the address information.
AKAs	List<EntityAKA>	The EntityAKA class contains the alternative name information.
EntityType	ListEntityType	This property contains the entity type. Possible values: - None - Business - Individual - Unknown - Vessel None is not valid in a request.
Gender	GenderType	This property contains the entity gender type. Possible values: - None - Female - Male - Unknown None is not valid in a request.
IDs	List<EntityID>	The EntityID class contains the entity ID information.

Class Properties (Continued)

Name	Type	Description
MatchInfo	List<EntityMatchInfo>	The EntityMatchInfo class contains information on the list entity that matched your input information.
Name	EntityName	The EntityName class contains the name information.
Phones	List<EntityPhone>	The EntityPhone class contains the phone number information.

Related Information

[EntityAdditionalInfo Class](#) on page 174

[EntityAddress Class](#) on page 174

[EntityAKA Class](#) on page 175

[EntityID Class](#) on page 176

[EntityMatchInfo Class](#) on page 143

[EntityName Class](#) on page 176

[EntityPhone Class](#) on page 177

EntityMatchInfo Class

The EntityMatchInfo class contains information on the list entity that matched your input information. Only accept list records contain this information.

Class Properties

Name	Type	Description
Best DOB	String	This property contains the DOB (date of birth) that most closely matched the input information.
BestName	String	This property contains the name that most closely matched the input information.
Checksum	Integer	This property contains a system-generated number that is used by XG to determine if True Match records have changed.
CustomFile	Boolean	True indicates that the list is a custom screening list. The default value is false.
EntityName	String	This property contains the entity name.
EntityScore	Integer	This property contains the match confidence score (0–100) that XG generated to indicate how closely the list record matched the input entity information. If a valid score is not available, –1 is returned.
EntityUniqueID	String	This property contains the entity's unique ID.
FileDate	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that the source agency provided the list that contained the match.

Class Properties (Continued)

Name	Type	Description
FileName	String	This property contains the name of the list that contained the match.
ID	Long	This property contains a system-generated number that identifies the entry in the record.
LastAlertResultID	Long	This property contains the last alert result ID.
MatchDate	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that XG identified the match.
MatchState	MatchStates	This property contains the state of the match. Possible values: - None - Baselined This value indicates that XG has added the necessary information for false match updates. - Realerted This value indicates that XG has identified a false match update.
OriginatingResultID	Long	This property contains the originating result ID.
PublishedDate	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that LexisNexis provided the screening list that contained the match.

ListRecord Class

The ListRecord class contains the record information when you add a record, update a record, or view record information. The record can either contain country information or entity information, not both.

When you update a record, the properties that are not included are deleted or set to false.

Class Properties

Name	Type	Description
Comments	String	This property contains the user-defined comments.
Country	CountryInfo	The CountryInfo class contains the country information when you add a record, update a record, or view record information. The property is required if the Entity property is not provided. If this property is included, the class cannot contain the Entity property.
DateListed	Date	The Date class contains date information.

Class Properties (Continued)

Name	Type	Description
Entity	EntityInfo	The EntityInfo class contains the entity information when you add a record, update a record, or view record information. This property is required if the Country property is not provided. If this property is included, the class cannot contain the Country property.
ID	Long	This property contains a system-generated number that identifies the record.
LastModified	DateTime	This property contains the date (YYYY-MM-DD) and 24-hour local time (HH:MM:SS) that the record was last modified.
ListID	Long	This property contains a system-generated primary key that identifies the list that contains the record. This property is required when you add a record to a list.
ReasonListed	String	The reason the record was added to the list. You use this property only for custom screening lists.
ReferenceNumber	String	This property contains a user-defined reference value.

Related Information

[CountryInfo Class](#) on page 140

[Date Class](#) on page 95

[EntityInfo Class](#) on page 142

ListRecordCriteria Class

The ListRecordCriteria class contains the criteria that records must meet to be returned in a list query. If multiple properties are included, the list must meet all criteria to be returned.

Class Properties

Name	Type	Description
Country	CountryCriteria	The CountryCriteria class contains the criteria a country record must meet to be returned in a list query.
DateListedEnd	Date	This property contains the latest date that the record was added to the list. The Date class contains date information.
DateListedStart	Date	This property contains the earliest date that the record was added to the list. The Date class contains date information.

Class Properties (Continued)

Name	Type	Description
Entity	EntityCriteria	The EntityCriteria class contains the criteria an entity record must meet to be returned in a list query. If the Country property is also provided, XG ignores this property.
ReasonListed	String	This property contains the reason that the record was added to the list. You use this property only for custom screening lists.
Reference	String	This property contains user-defined reference information.

Related Information

[CountryCriteria Class](#) on page 139

[Date Class](#) on page 95

[EntityCriteria Class](#) on page 141

ResultInfo Class

The ResultInfo class contains the alert information that XG used to create the list record.

Only business entities or individual entities can be added to lists. When you add a record to an accept list, the record must contain a complete address or ID number.

Class Properties

Name	Type	Description
NewAlertState	AlertStateType	This property contains the alert state. Possible values: - None - Closed - Open - Unassigned
NewMatchStates	List<WatchlistMatchState>	The WatchlistMatchState class contains the decision information if separate decisions were set for each list match in the alert. This property is required if the alert is configured to disposition each list match separately.
NewStatus	String	This property contains the new alert status.
Note	String	This property contains a user-defined note. This property is required if a note requirement was configured in the XG application and the action that required a note was performed.

Class Properties (Continued)

Name	Type	Description
Record	ResultRecord	The ResultRecord class contains the alert that XG generated for the search. This property is required if the ResultID property is not provided.  <i>XG ignores information that is not relevant to list searching. For example, due diligence results are not added to the list record.</i>
ResultID	Long	This property contains a system-generated number that identifies the alert, to be added to the list, in the XG database. This property is required if the Record property is not provided. If the Record property is included, this property is ignored.

Related Information

[WatchlistMatchState Class](#) on page 67

[ResultRecord Class](#) on page 62

SearchRecordResults Class

The SearchRecordResults class contains the record IDs of the alerts that met the criteria in a result query.

Class Properties

Name	Type	Description
RecordIDs	List<Long>	This property contains a list of record IDs that identify the records that were returned.
TotalNumRecords	Long	This property contains the total number of records that were returned.

Chapter 7

EFT Parsed Input Classes

With the Search method in the Search interface, input EFT (electronic funds transfer) data, or payment transaction data, is searched as a text string that is provided in the InputEFT class. LexisNexis® Bridger Insight® XG (XG) web services return the input payment transaction data as structured information when a potential list match is reported.

Related Information

[Search Method](#) on page 28

[InputEFT Class](#) on page 52

ACH Parsed Input Classes

Search requests for ACH (Automated Clearing House) records include various input classes. These classes are returned in the list screening results of the Search interface and the result queries that are performed in the Results interface.

Related Information

[List Screening Result Classes](#) on page 79

[Result Management Classes](#) on page 102

ParsedACH Class

The ParsedACH class contains the input information.

For information on record content and format, see <http://www.nacha.org>.

Class Properties

Name	Type	Description
Notes	List<ParsedACHNote>	The ParsedACHNote class contains the input note information.
Parties	List<ParsedACHParty>	The ParsedACHParty class contains the input party information.

Related Information

[ParsedACHField Class](#) on page 150

[ParsedACHNote Class](#) on page 150

[ParsedACHParty Class](#) on page 151

ParsedACHAdditionalInfo Class

The ParsedACHAdditionalInfo class contains the additional information for the input party.

Class Properties

Name	Type	Description
Field	Integer	This property contains the field number. This property is required.
Info	String	This property contains the additional information.
RecordTypeCode	Integer	This property contains the record type code. This property is required.

ParsedACHAddress Class

The ParsedACHAddress class contains the address information for the input party.

Class Properties

Name	Type	Description
City	String	This property contains the city.
CityStateProvince	String	This property contains the city, the state, or the province.
Country	String	This property contains the country.
CountryPostalCode	String	This property contains the country postal code.
PostalCode	String	This property contains the postal code.
StateProvince	String	This property contains the state or the province.
StreetAddress	String	This property contains the street address.
Type	AddressType	This property contains the address type. Possible values: • None • Current • Mailing • Previous • Unknown None is not valid in a request.

ParsedACHField Class

The ParsedACHField class contains the field information.

Class Properties

Name	Type	Description
AddendaTypeCode	String	This property contains the addenda type code.
Contested	Boolean	True indicates contested.
Dishonored	Boolean	True indicates dishonored.
Field	Integer	This property contains the field number.
RecordTypeCode	Integer	This property contains the record type code.
Refused	Boolean	True indicates refused.
Value	String	This property contains the field value.

ParsedACHID Class

The ParsedACHID class contains the ID information for the input party.

Class Properties

Name	Type	Description
Number	String	This property contains the ID number.
Type	IDType	This property contains the ID type.

ParsedACHNote Class

The ParsedACHNote class contains the input note information.

Class Properties

Name	Type	Description
AdditionalTextSearch	Boolean	True indicates that XG was set to search all fields for one or more of the following data types: BIC (Business Identifier Code) values, country codes, country names, or names.
EDITRNDDataSegment Identifier	String	This property contains the edit RN data segment identifier.
Field	Integer	This property contains the field.
IDField	Integer	This property contains the ID field.

Class Properties (Continued)

Name	Type	Description
IDRecordTypeCode	Integer	This property contains the ID record type code. This property is required.
Note	String	This property contains the note text.
OriginalReceivingDFI Identification	String	This property contains the original receiving depository financial institution identification.
OriginatingCompanyID	String	This property contains the originating company ID.
RecordTypeCode	Integer	This property contains the record type code. This property is required.
ReferenceID1	String	This property contains the reference ID field.
ReferenceID2	String	This property contains the reference ID field.
SequenceNumber	String	This property contains the sequence number.
TraceTypeCode	String	This property contains the trace type code.
TypeCode	String	This property contains the type code.

ParsedACHParty Class

The ParsedACHParty class contains the input party information.

Class Properties

Name	Type	Description
AdditionalInfo	List<String>	This property contains user-defined additional information.
Addresses	List<ParsedACHAddress>	The ParsedACHAddress class contains the address information for the input party.
IDs	List<ParsedACHID>	The ParsedACHID class contains the ID information for the input party.
Name	String	This property contains the party name.
Type	EFTPartyType	This property contains the EFT party type for payment transaction records.

Related Information

[ParsedACHAdditionalInfo Class](#) on page 149

[ParsedACHAddress Class](#) on page 149

[ParsedACHID Class](#) on page 150

[EFT Party Types](#) on page 152

EFT Party Types

Payment transaction data can include EFT party types.

The following EFT party types are available:

- AccountCredited
- AccountDebited
- AccountIdentification
- AccountServicingFI
- AccountWithInstitution
- AdviseThroughFI
- Applicant
- ApplicantFI
- ApplicationHeaderBlockParty
- AvailableWithFI
- BasicHeaderBlockParty
- Beneficiary
- BeneficiaryFI
- BeneficiaryOfCommodity
- Borrower
- Broker
- CalculationAgent
- CentralCounterpartyClearingHouse
- ChangedApplicant
- ChargesAccount
- ChargesAccountAgent
- ChequeDeliverTo
- ChequeFrom
- ClearingBroker
- ClearingExceptionParty
- Contact
- CorrespondentFI
- Counterparty
- CountryofDomicile
- CreditMandateSigner
- Creditor
- CreditorAccount
- CreditorAgent
- CreditorAgentAccount
- CreditorsFI
- DealingBranchPartyA
- DealingBranchPartyB
- DebitInstitution
- Debtor
- DebtorAccount
- DebtorAgent
- DebtorAgentAccount
- DebtorsFI
- DelivererOfCommodity
- DeliveryAgent
- DepositoryofCSD
- DocumentIssuer
- Drawee
- DrawerFI
- EnquiredParty
- ExecutingBroker
- ExecutionVenue
- ForwardingAgent
- Fund
- FundManager
- FundOrBeneficiaryCustomer
- FundOrInstructingParty
- FXCounterparty
- InitiatingInstitution
- InitiatingParty
- InstructingFI
- InstructingParty
- IntermediaryAgent
- IntermediaryAgentAccount
- IntermediaryFI
- Investor
- Invoicee
- Invoicer

- Issuer
- IssuingFI
- LoanDepositCounterparty
- MessageOriginator
- NegotiatingFI
- None
- Offeror
- OrderingCustomer
- OrderingFI
- OriginalCreditMandateSigner
- OriginalCreditorAgent
- OriginalCreditorAgentAccount
- OriginalDebtor
- OriginalDebtorAccount
- OriginalDebtorAgent
- OriginalDebtorAgentAccount
- OriginalReceivingFI
- OriginationFI
- Originator
- PartyA
- PartyB
- Payee
- Payer
- PrimaryDocumentIssuer
- Principal
- Receiver
- ReceiverOfCommodity
- ReceiversCorrespondent
- ReceivingAgent
- ReceivingFI
- ReimbursingFI
- RelatedParty
- Remittance
- RemittingAgent
- ReportingParty
- RequestedConfirmation
- SafekeepingAccount
- SecondaryDocumentIssuer
- SellingAgent
- SenderOfCashLetter
- SendersCorrespondent
- SendingFI
- SSIRecipient
- SubmittingParty
- ThirdReimbursementFI
- TradeParty
- TransferAgent
- UltimateCreditor
- UltimateDebtor

Fedwire Parsed Input Classes

Search requests for Fedwire records include various input classes. These classes are returned in the list screening results of the Search interface and the result queries that are performed in the Results interface.

Related Information

[List Screening Result Classes](#) on page 79

[Result Management Classes](#) on page 102

ParsedFedwire Class

The ParsedFedwire class contains the Fedwire input information.

For information on data content and record format, see: <http://www.frbervices.org>.

Class Properties

Name	Type	Description
Notes	List <ParsedFedwire Note>	The ParsedFedwireNote class contains the input note information.
Parties	List <ParsedFedwire Party>	The ParsedFedwireParty class contains the input party information.

Related Information

[ParsedFedwireField Class](#) on page 156

[ParsedFedwireNote Class](#) on page 156

[ParsedFedwireParty Class](#) on page 157

ParsedFedwireAdditionalInfo Class

The ParsedFedwireAdditionalInfo class contains the additional information for the input party.

Class Properties

Name	Type	Description
Country	String	This property contains the country.
Info	String	This property contains additional input information.
ParsedInfo	String	This property contains additional structured input information.
Type	AdditionalInfoType	This property contains the information type.

Related Information

[Possible Additional Information Types](#) on page 180

ParsedFedwireAddress Class

The ParsedFedwireAddress class contains the address information for the input party.

Class Properties

Name	Type	Description
AddressLine2	String	This property contains the line 2 of the address.
AddressLine3	String	This property contains the line 3 of the address.
AddressLine4	String	This property contains the line 4 of the address.
AddressLine5	String	This property contains the line 5 of the address.
AddressLine6	String	This property contains the line 6 of the address.
AddressLine7	String	This property contains the line 7 of the address.
BuildingNumber	String	This property contains the building number.
City	String	This property contains the city.
Country	String	This property contains the country.
Department	String	This property contains the department.
PostalCode	String	This property contains the postal code.
StateProvince	String	This property contains the state or the province.
StreetName	String	This property contains the street name.
SubDepartment	String	This property contains the sub-department.
Type	AddressType	This property contains the address type. Possible values: • None • Current • Mailing • Previous • Unknown None is not valid in a request. This property is required.

ParsedFedwireAmount Class

The ParsedFedwireAmount class contains the input transaction amount information.

Class Properties

Name	Type	Description
Amount	String	This property contains the transaction amount.
CurrencyCode	String	This property contains the currency code.

ParsedFedwireField Class

The ParsedFedwireField class contains the Fedwire field information.

Class Properties

Name	Type	Description
Amount	ParsedFedwireAmount	The ParsedFedwireAmount class contains the input transaction amount information.
Element	Integer	This property contains the element number.
Tag	String	This property contains the field tag.
Value	String	This property contains the field value.

Related Information

[ParsedFedwireAmount Class](#) on page 155

ParsedFedwireID Class

The ParsedFedwireID class contains the input party ID information.

Class Properties

Name	Type	Description
CountryCode	String	This property contains the country code.
Issuer	String	This property contains the issuer of the ID.
Number	String	This property contains the ID number.
Type	IDType	This property contains the ID type. This property is required.

Related Information

[Possible ID Types](#) on page 180

ParsedFedwireNote Class

The ParsedFedwireNote class contains the input note information.

Class Properties

Name	Type	Description
Note	String	This property contains the remittance note.
Tag	String	This property contains the remittance note tag.

ParsedFedwireParty Class

The ParsedFedwireParty class contains the input party information.

Class Properties

Name	Type	Description
AdditionalInfo	List<ParsedFedwireAdditionalInfo>	The ParsedFedwireAdditionalInfo class contains the additional information for the input party.
Addresses	List<ParsedFedwireAddress>	The ParsedFedwireAddress class contains the address information for the input party.
EntityType	ResultEntityType	This property contains the entity type. Possible values: • Business • Individual • Text • Unknown • Unstructured
Fields	List<ParsedFedwireNote>	The ParsedFedwireNote class contains the input note information.
IDs	List<ParsedFedwireID>	The ParsedFedwireID class contains the input party ID information.
Name	String	This property contains the Fedwire party name.
Phones	List<ParsedFedwirePhone>	The ParsedFedwirePhone class contains the phone information for the input party.
Type	EFTPartyType	This property contains the EFT party type for payment transaction records.

Related Information

[ParsedFedwireAdditionalInfo Class](#) on page 154

[ParsedFedwireAddress Class](#) on page 154

[ParsedFedwireID Class](#) on page 156

[ParsedFedwirePhone Class](#) on page 158

[EFT Party Types](#) on page 152

[ParsedFedwireNote Class](#) on page 156

ParsedFedwirePhone Class

The ParsedFedwirePhone class contains the phone information for the input party.

Class Properties

Name	Type	Description
Number	String	This property contains the phone number.
Type	PhoneType	This property contains the phone type. Possible values: • Business • Cell • Fax • Home • Work • Unknown • None None is not valid in a request. This property is required.

ISO 20022 Parsed Input Classes

Search requests for ISO (International Organization for Standardization) 20022 records include various input classes. These classes are returned in the list screening results of the Search interface and the result queries that are performed in the Results interface.

Related Information

[List Screening Result Classes](#) on page 79

[Result Management Classes](#) on page 102

ParsedISO20022 Class

The ParsedISO20022 class contains the ISO 20022 input information.

For information on record content and format, see: <http://www.iso20022.org>.

Class Properties

Name	Type	Description
Notes	List<ParsedISO20022Note>	The ParsedISO20022Note class contains the input note information.
Parties	List<ParsedISO20022Party>	The ParsedISO20022Party class contains the input party information.

Related Information

[ParsedISO20022Field Class](#) on page 160

[ParsedISO20022Note Class](#) on page 161

[ParsedISO20022Party Class](#) on page 161

ParsedISO20022AdditionalInfo Class

The ParsedISO20022AdditionalInfo class contains the additional information for the input party.

Class Properties

Name	Type	Description
CountryOfBirth	String	This property contains the country of birth.
Info	String	This property contains additional input information.
ParsedInfo	String	This property contains additional structured input information.
ProvinceOfBirth	String	This property contains the province of birth.
Type	AdditionalInfoType	This property contains the information type.

Related Information

[Possible Additional Information Types](#) on page 180

ParsedISO20022Address Class

The ParsedISO20022Address class contains the address information for the input party.

Class Properties

Name	Type	Description
BuildingNumber	String	This property contains the building number.
City	String	This property contains the city.
Country	String	This property contains the country.

Class Properties (Continued)

Name	Type	Description
Department	String	This property contains the department.
Lines	List<String>	This property contains the address lines.
PostalCode	String	This property contains the postal code.
StateProvince	String	This property contains the state or the province.
StreetName	String	This property contains the street name.
SubDepartment	String	This property contains the sub-department.
Type	AddressType	This property contains the address type. Possible values: • None • Current • Mailing • Previous • Unknown None is not valid in a request. This property is required.

ParsedISO20022Field Class

The ParsedISO20022Field class contains the ISO 20022 field information.

Class Properties

Name	Type	Description
CurrencyCode	Boolean	This property contains the currency code. This property is required.
Value	String	This property contains the field value.
XPath	String	This property contains the XPath value.

ParsedISO20022ID Class

The ParsedISO20022ID class contains the ID information for the input party.

Class Properties

Name	Type	Description
Issuer	String	This property contains the issuer of the ID.
Label	String	This property contains the ID label.

Class Properties (Continued)

Name	Type	Description
Number	String	This property contains the ID number.
Type	IDType	This property contains the ID type.

Related Information

[Possible ID Types](#) on page 180

ParsedISO20022Note Class

The ParsedISO20022Note class contains the input note information.

Class Properties

Name	Type	Description
Note	String	This property contains the note text.
Tag	String	This property contains the note tag.

ParsedISO20022Party Class

The ParsedISO20022Party class contains the input party information.

Class Properties

Name	Type	Description
AdditionalInfo	List<ParsedISO20022AdditionalInfo>	The ParsedISO20022AdditionalInfo class contains the additional information for the input party.
Addresses	List<ParsedISO20022Address>	The ParsedISO20022Address class contains the address information for the input party.
EntityType	ResultEntityType	This property contains the entity type. Possible values: - Business - Individual - Text - Unknown - Unstructured
IDs	List<ParsedISO20022ID>	The ParsedISO20022ID class contains the ID information for the input party.
Name	String	This property contains the party name.

Class Properties (Continued)

Name	Type	Description
Notes	List<ParsedISO20022Note>	The ParsedISO20022Note class contains the input note information.
Phones	List<ParsedISO20022Phone>	The ParsedISO20022Phone class contains phone information for the input party.
Type	EFTPartyType	The EFT party type.

Related Information

[ParsedISO20022AdditionalInfo Class](#) on page 159

[ParsedISO20022Address Class](#) on page 159

[ParsedISO20022ID Class](#) on page 160

[ParsedISO20022Note Class](#) on page 161

[ParsedISO20022Phone Class](#) on page 162

[EFT Party Types](#) on page 152

ParsedISO20022Phone Class

The ParsedISO20022Phone class contains phone information for the input party.

Class Properties

Name	Type	Description
Number	String	This property contains the phone number.
Type	PhoneType	This property contains the phone type. Possible values: • Business • Cell • Fax • Home • Work • Unknown • None None is not valid in a request. This property is required.

SWIFT Parsed Input Classes

Search requests for SWIFT (Society for Worldwide Interbank Financial Telecommunication) records include various input classes. These classes are returned in the list screening results of the Search interface and the result queries that are performed in the Results interface.

Related Information

[List Screening Result Classes](#) on page 79

[Result Management Classes](#) on page 102

ParsedSWIFT Class

The ParsedSWIFT class contains the SWIFT input information. For information on record content and format, see: <http://www.swift.com>.



*SWIFT formats that were supported prior to the XG web services 4.4.5 release return input values in both the **Fields** array and fields such as **Amount**. SWIFT formats that were released after the XG web services 4.4.5 release return input values in only the **Fields** array.*

Class Properties

Name	Type	Description
Fields	List<ParsedSWIFTField>	The ParsedSWIFTField class contains the SWIFT field information.
MessageType	String	This property contains the message type.
Notes	List<ParsedSWIFTNote>	The ParsedSWIFTNote class contains the input note information.
Parties	List<ParsedSWIFTParty>	The ParsedSWIFTParty class contains the input party information.

Related Information

[ParsedSWIFTAmount Class](#) on page 165

[ParsedSWIFTField Class](#) on page 165

[ParsedSWIFTNote Class](#) on page 166

[ParsedSWIFTParty Class](#) on page 167

ParsedSWIFTAdditionalInfo Class

The ParsedSWIFTAdditionalInfo class contains the additional information for the input party.

Class Properties

Name	Type	Description
Country	String	This property contains the country.
Info	String	This property contains the additional information.
ParsedInfo	String	This property contains the structured additional information.
Type	AdditionalInfoType	This property contains the additional information type.

Related Information

[Possible Additional Information Types](#) on page 180

ParsedSWIFTAddress Class

The ParsedSWIFTAddress class contains the address information for the input party.

Class Properties

Name	Type	Description
AddressLine1	String	This property contains the line 1 of the address.
AddressLine2	String	This property contains the line 2 of the address.
AddressLine3	String	This property contains the line 3 of the address.
AddressLine4	String	This property contains the line 4 of the address.
City	String	This property contains the city.
Country	String	This property contains the country.
Type	AddressType	This property contains the address type. Possible values: • None • Current • Mailing • Previous • Unknown None is not valid in a request. This property is required.

ParsedSWIFTAmount Class

The ParsedSWIFTAmount class contains the input transaction amount information.

Class Properties

Name	Type	Description
AdditionalInformation	String	This property contains additional input information.
Amount	String	This property contains the transaction amount.
Code	String	This property contains the transaction code.
CurrencyCode	String	This property contains the currency code.
Date	String	This property contains the transaction date.
DebitCreditMark	String	This property contains the debit or the credit mark.
MaturityPeriodType	String	This property contains the maturity period type.
Number	String	This property contains the transaction number.
NumberOfDaysOrMonths	String	This property contains the number of days or months.
Type	String	This property contains the transaction type.
Type2	String	This property contains the transaction type.
Units	String	This property contains the units.

ParsedSWIFTField Class

The ParsedSWIFTField class contains the SWIFT field information.

Class Properties

Name	Type	Description
Amount	ParsedSWIFTAmount	The ParsedSWIFTAmount class contains the input transaction amount information.
Qualifier	String	This property contains the qualifier.
Sequence	String	This property contains the sequence.
SubSequence	String	This property contains the sub-sequence.
Tag	String	This property contains the tag value.
Value	String	This property contains the value field.

Related Information

[ParsedSWIFTAmount Class](#) on page 165

[ParsedSWIFTOriginalMessage Class](#) on page 166

ParsedSWIFTID Class

The ParsedSWIFTID class contains the ID information for the input party.

Class Properties

Name	Type	Description
CountryCode	String	This property contains the ID country code.
Issuer	String	This property contains the issuer of the ID.
Number	String	This property contains the ID number.
Tag	String	This property contains the ID tag.
Type	IDType	This property contains the ID type. This property is required.

Related Information

[Possible ID Types](#) on page 180

ParsedSWIFTNote Class

The ParsedSWIFTNote class contains the input note information.

Class Properties

Name	Type	Description
Note	String	This property contains the note text.
Tag	String	This property contains the note tag.

ParsedSWIFTOriginalMessage Class

The ParsedSWIFTOriginalMessage class contains the input original message details.

Class Properties

Name	Type	Description
Date	String	This property contains the date.
MessageType	String	This property contains the message type.
SessionNumberAndISN	String	This property contains the session number and the ISN (input sequence number).

ParsedSWIFTParty Class

The ParsedSWIFTParty class contains the input party information.

Class Properties

Name	Type	Description
AdditionalInfo	List<ParsedSWIFTAdditionalInfo>	The ParsedSWIFTAdditionalInfo class contains the additional information for the input party.
Addresses	List<ParsedSWIFTAddress>	The ParsedSWIFTAddress class contains the address information for the input party.
IDs	List<ParsedSWIFTID>	The ParsedSWIFTID class contains the ID information for the input party.
Name	String	This property contains the party name.
Tag	String	This property contains the party tag.
Type	EFTPartyType	This property contains the EFT party type for payment transaction records.

Related Information

[ParsedSWIFTAdditionalInfo Class](#) on page 164

[ParsedSWIFTAddress Class](#) on page 164

[ParsedSWIFTID Class](#) on page 166

[EFT Party Types](#) on page 152

Chapter 8

Entity Information

LexisNexis® Bridger Insight® XG (XG) web services include entity information in a number of methods that are available in the Search interface, the Results interface, or the List Maintenance interface.

Input Entity Classes

With the Search method in the Search interface, the input entity classes can be provided in the request or returned in the response. XG can also return this information from result queries that are performed in the Results interface.

Related Information

[Search Input Classes](#) on page 51

[Record Information Classes](#) on page 60

[Result Management Classes](#) on page 102

Date Class

The Date class contains date information.

Class Properties

Name	Type	Description
Day	Integer	This property contains the day value in a structured date.
Month	Integer	This property contains the month value in a structured date.
Unparsed	String	This property contains the unstructured date value.
Year	Integer	This property contains the year value in a structured date.

InputAdditionalInfo Class

The InputAdditionalInfo class contains the additional information that helps identify the input entity.

Class Properties

Name	Type	Description
Date	Date	The Date class contains date information.
Label	String	This property contains a user-defined label.
Type	AdditionalInfoType	This property contains the additional information type. This property is required.
Value	String	This property contains the user-defined input value.

Related Information

[Date Class](#) on page 95

[Possible Additional Information Types](#) on page 180

InputAddress Class

The InputAddress class contains the input address information.

Class Properties

Name	Type	Description
BuildingName	String	This property contains the building name.
BuildingOrStreet Number	String	This property contains the building number or the street number.
City	String	This property contains the city or the town.
Country	String	This property contains the country.
County	String	This property contains the county.
District	String	This property contains the district.
FloorNumber	String	This property contains the floor number.
FullAddress	String	This property contains the full address. XG ignores information that is provided in this property if enough information is provided in the structured address properties.
HouseNumber	String	This property contains the house number.
IntlStreetType	String	This property contains the international street type.
PostalCode	String	This property contains the postal code or the area.
StateProvinceDistrict	String	This property contains the state, the province, or the district.

Class Properties (Continued)

Name	Type	Description
Street1	String	This property contains the line 1 of the address. This property may contain unstructured address information.
Street2	String	This property contains the line 2 of the address. This property may contain unstructured address information.
StreetName	String	This property contains the street name.
StreetPostDirection	String	This property contains the street direction after the street name. For example, this property would contain W if the address is 123 Main St W.
StreetPreDirection	String	This property contains the street direction before the street name. For example, this property would contain N if the address is 123 N Main St.
StreetSuffixOrType	String	This property contains the street type. For example: Avenue, Drive, Street, Boulevard.
Suburb	String	This property contains the suburb.
Type	AddressType	This property contains the address type. Possible values: • None • Current • Mailing • Previous • Unknown None is not valid in a request. This property is required.
UnitDesignation	String	This property contains the unit designation. Examples of unit designation include: unit, suite, apartment, or flat number.
UnitNumber	String	This property contains the unit number.

InputID Class

The InputID class contains the input ID information. The input ID may come from either an **InputEntity** property search, an **InputText** property search or an **InputBasicID** property search.

Class Properties

Name	Type	Description
AccountType	String	This property contains the account type.
CityOfIssue	String	This property contains the city where the ID was issued.
CountryOfBirth	String	This property contains the country where the person was born.
CountyOfIssue	String	This property contains the county where the ID was issued.
DateExpires	Date	This property contains the date the ID expires. The Date class contains date information.
DateIssued	Date	This property contains the date that the ID was issued. The Date class contains date information.
DistrictOfIssue	String	This property contains the district where the ID was issued.
DriverLicenceVersionNumber	String	This property contains the driver license version number.
FamilyNameAtBirth	String	This property contains the family name of the person at birth.
FamilyNameAtCitizenship	String	This property contains the family name of the person at the time of citizenship.
Issuer	String	This property contains the state or the country that issued the ID.
Label	String	This property contains a user-defined ID label. You can view this label in the XG application only if the ID Type equals Other.
MachineReadableLine1	String	This property contains a machine-readable line typically used for passports or visas.
MachineReadableLine2	String	This property contains a machine-readable line typically used for passports or visas.
Number	String	This property contains the ID number.
PlaceOfBirth	String	This property contains the place where the person was born.
ProvinceOfIssue	String	This property contains the province where the ID was issued.
StateOfBirth	String	This property contains the state where the person was born.
Type	IDType	This property contains the ID type. This property is required.

Related Information

[Date Class](#) on page 95

[Possible ID Types](#) on page 180

InputName Class

The InputName class contains the input name information. This class is required for all Search requests.

For individual entities, the **First** property and the **Last** property are required or the **Full** property is required. For business entities or unknown entities, the **Full** property is required.



*Records that contain the **Full** property can only be searched against screening lists.*

All records within a request must contain the same properties.

Class Properties

Name	Type	Description
First	String	This property contains the first name. You use this property only for individual entities.
Full	String	This property contains the full name.
Generation	String	This property contains the generation. Examples of generation information include Junior or III. You use this property only for individual entities.
Last	String	This property contains the last name. You use this property only for individual entities.
MaidenName	String	This property contains the maiden name. You use this property only for individual entities.
Middle	String	This property contains the middle name. You use this property only for individual entities.
SecondSurname	String	This property contains the second surname. You use this property only for individual entities.
Title	String	This property contains the title. You use this property only for individual entities.

InputPhone Class

The InputPhone class contains the input phone number information.

Class Properties

Name	Type	Description
Number	String	This property contains the phone number.
Type	PhoneType	This property contains the phone type. Possible values: • Business • Cell • Fax • Home • Work • Unknown • None None is not valid in a request. This property is required.

Entity Record Classes

With the Search method in the Search interface, the entity record classes can be returned in the list screening results. XG can also return this information in list queries that are performed in the List Maintenance interface.

Related Information

[List Screening Result Classes](#) on page 79

[List Record Management Classes](#) on page 136

Date Class

The Date class contains date information.

Class Properties

Name	Type	Description
Day	Integer	This property contains the day value in a structured date.
Month	Integer	This property contains the month value in a structured date.
Unparsed	String	This property contains the unstructured date value.
Year	Integer	This property contains the year value in a structured date.

EntityAdditionalInfo Class

The EntityAdditionalInfo class contains the additional information that helps identify the list entity.

Class Properties

Name	Type	Description
Comments	String	This property contains additional user-defined comments.
ID	Long	This property contains a system-generated number that identifies the entry in the record.
Type	AdditionalInfoType	This property contains the additional information type. This property is required.
Value	String	This property contains the additional information.

Related Information

[Possible Additional Information Types](#) on page 180

EntityAddress Class

The EntityAddress class contains the address information.

Class Properties

Name	Type	Description
City	String	This property contains the city.
Comments	String	This property contains any additional comments. For example, this property may include a date for when the address was current.
Country	String	This property contains the country.
ID	Long	This property contains a system-generated number that identifies the entry in the record.
PostalCode	String	This property contains the postal code.
StateProvinceDistrict	String	This property contains the state, the province, or the district.
Street1	String	This property contains the line 1 of the address.

Class Properties (Continued)

Name	Type	Description
Street2	String	This property contains the line 2 of the address.
Type	AddressType	This property contains the address type. Possible values: • None • Current • Mailing • Previous • Unknown None is not valid in a request. This property is required.

EntityAKA Class

The EntityAKA class contains the alternative name information.

Class Properties

Name	Type	Description
Category	AKACategoryType	This property contains the AKA (also known as) category type. This type indicates the level that the alternative name is associated with the entity. Possible values: • None • Strong • Weak
Comments	String	This property contains additional AKA information. For example, this property may include the DOB (date of birth) information or the POB (place of birth) information for the AKA entity.
ID	Long	This property contains a system-generated number that identifies the entry in the record.
Name	EntityName	The EntityName class contains the name information.
Type	AKAType	This property contains the alternative name type. Possible values: • None • AKA (also known as) • FKA (formerly known as) • NKA (now known as)

Related Information

[EntityName Class](#) on page 176

EntityID Class

The EntityID class contains the entity ID information.

Class Properties

Name	Type	Description
Comments	String	This property contains user-defined comments about the ID.
DateExpires	Date	The Date class contains date information.
DateIssued	Date	The Date class contains date information.
Issuer	String	This property contains the state or the country that issued the ID.
ID	Long	This property contains a system-generated number that identifies the entry in the record.
Label	String	This property contains the ID label.
Number	String	This property contains the ID number.
Type	IDType	This property contains the ID type. This property is required.

Related Information

[Date Class](#) on page 95

[Possible ID Types](#) on page 180

EntityName Class

The EntityName class contains the name information.

Class Properties

Name	Type	Description
First	String	This property contains the first name.
Full	String	This property contains the full name.
Generation	String	This property contains the generation. Examples of generation information include Junior or III.

Class Properties (Continued)

Name	Type	Description
Last	String	This property contains the last name.
Middle	String	This property contains the middle name.
Title	String	This property contains the title.

EntityPhone Class

The EntityPhone class contains the phone number information.

Class Properties

Name	Type	Description
Comments	String	This property contains any additional comments about the phone information.
ID	Long	This property contains a system-generated number that identifies the entry in the record.
Number	String	This property contains the phone number.
Type	PhoneType	This property contains the phone type. Possible values: • Business • Cell • Fax • Home • Work • Unknown • None None is not valid in a request. This property is required.

Entity Criteria Classes

The entity criteria classes can be provided in the result queries that are performed in the Results interface or in the list record queries that are performed in the List Maintenance interface.

Related Information

[Result Management Classes](#) on page 102

[List Record Management Classes](#) on page 136

EntityCriteriaAdditionalInfo Class

The EntityCriteriaAdditionalInfo class contains the additional information that the record must meet to be returned.

If only one of the **Value1** or **Value2** properties is provided, the record information must meet that value. For example, you set the **Type** to **Citizenship** and provide **United States** in the **Value1** property. Only records that contain a **Citizenship** of **United States** are returned.

If both properties are provided, the record information must meet a value within the provided range. For example, you set the **Type** to **DOB** and provide **01/01/1955** in the **Value1** property and **01/01/1965** for the **Value2** property. Only records that contain a **DOB** within the range of **01/01/1955–01/01/1965** are returned.

Class Properties

Name	Type	Description
Type	AdditionalInfoType	This property contains the additional information type. This property is required.
Value1	String	This property contains additional information that is contained in the record, or the start of the range that includes the additional information.
Value2	String	This property contains additional information that is contained in the record, or the end of the range that includes the additional information.

Related Information

[Possible Additional Information Types](#) on page 180

EntityCriteriaAddress Class

The EntityCriteriaAddress class contains the address information and the phone information that the record must meet to be returned.

Class Properties

Name	Type	Description
BuildingName	String	This property contains the building name.
City	String	This property contains the city.
Country	String	This property contains the country.
District	String	This property contains the district.
FloorNumber	String	This property contains the floor number.

Class Properties (Continued)

Name	Type	Description
Phone	Long	This property contains the phone number.
PostalCode	String	This property contains the postal code.
StateProvinceDistrict	String	This property contains the state, the province, or the district.
Street1	String	This property contains the line 1 of the address.
Street2	String	This property contains the line 2 of the address.
Suburb	String	This property contains the suburb.

EntityCriteriaID Class

The EntityCriteriaID class contains the ID information that the record must meet to be returned.

Class Properties

Name	Type	Description
Label	String	This property contains the ID label.
Number	String	This property contains the ID number.
Type	IDType	This property contains the ID type. This property is required.

Related Information

[Possible ID Types](#) on page 180

Possible Type Values

A number of classes contain additional information or IDs for entities in the Lists interface. These classes include a **Type** property that indicates the kind of information the class provides.

Possible Additional Information Types

Entity information can include various additional information types.

The following additional information type values are available:

- None (not valid in a request)
- Citizenship
- Complexion
- DistinguishingMarks
- DOB
- EyeColor
- HairColor
- Height
- Incident
- IPAddress
- MothersName
- Nationality
- Occupation
- Other
- PlaceOfBirth
- Position
- Race
- VesselCallSign
- VesselFlag
- VesselGRT
- VesselOwner
- VesselTonnage
- VesselType
- Weight

Possible ID Types

Entity information can include various ID types.

The following ID type values are available:

- None (not valid in a request or in a custom screening list for an entity unique ID)
- ABARouting
- Account
- AlienRegistration
- BankIdentifierCode
- BankPartyID
- Cedula
- ChipsUID
- CustomerNumber
- DriversLicense
- DUNS
- EFTCode
- EIN
- GLN
- IBAN
- IBEI
- MedicareID
- MedicareReference
- Member
- Military
- National
- NIT
- Other
- Passport
- ProprietaryUID
- ProviderID
- RTACardNumber
- SSN
- SwiftBEI
- SwiftBIC
- TaxID
- VISA

Chapter 9

Errors

LexisNexis® Bridger Insight® XG (XG) web services can return service fault errors that prevented the request from being completed and result errors that were returned from searches.

ServiceFault Class

The ServiceFault class returns any errors that were generated by the system or method.

Class Properties

Name	Type	Description
Message	String	This property contains the ServiceFault error message.
Type	ServiceFaultType	This property contains the ServiceFault type. Possible values: • AccessDenied • Error • FreeTrialLapsed • InvalidCredentials • MustChangePassword • PasswordExpired • UnsupportedVersion • UserDisabled • UserLocked • WatchlistsExpired

Related Information

[ServiceFault Error Messages](#) on page 181

ServiceFault Error Messages

ServiceFault errors contain the following text: The following error occurred while processing your request. Please contact your system administrator.

The errors also include the XG error message, a search message, or one of the messages in the following table:

ServiceFault Error Messages

Error Message
{0} is not an allowed assignment type for the given alert state: {1}
Access was denied
Alert Inbox privileges are required to assign results to role {0}
Alert Inbox privileges are required to assign results to user {0}
A list with the name {0} already exists
A list with the name {0} already exists as an accept list
A list with the name {0} already exists as an custom country watchlist
A list with the name {0} already exists as an custom watchlist
A list with the unique id {0} already exists
A list with the unique id {0} already exists as an accept list
A list with the unique id {0} already exists as an custom country watchlist
A list with the unique id {0} already exists as an custom watchlist
All match records must have a disposition set before the record can be added to the list
A note is required to add the entity to the list
A result assigned to division {0} cannot be added to the accept list {1}
A status is required to add the entity to the list
At least one provider search is required for a custom search configuration
At least one role id is required in the assigned to list
Cannot change the Windows password
Change password required
Configuration setting name is null or empty
Division is required to assign results
Division is required to assign results to user {0}
Entity has already been added to an accept list
Entity Type Unknown is not valid with due diligence providers
Failed to log message to event log: {0}\r\n\tOriginal message: {1}
Failed to log message to file: {0}\r\n\tOriginal message: {1}
Free trail has lapsed
Invalid Credentials
Login failed
Missing Client ID
Missing Password
Missing User ID
None of the specified roles are members of the division
None of the specified users are members of the division
Note is required when alert state changed
Note is required when record is reassigned
Note is required when record status is changed
One or more selected providers can only accept parsed names as valid input

ServiceFault Error Messages (Continued)

Error Message
One user id is required in the assigned to list
Password change failed
Password has expired
Result id or result information is required to add a record to a list
Results cannot be assigned since there are no roles defined
Some of the input records are missing record ids
Some of the input records have the same record id
Status is required when closing an alert
Status is required when reassigning an alert
The {0} provider is not loaded into the system
The AddRecord method is not allowed with accept lists
The AddRecord method is not allowed with custom country watchlists
The array of result record ids is null or empty
The array of run ids is null or empty
The assignment information is invalid
The assignment type Unknown is invalid
The attachment id is invalid: {0}
The attachment information object is null
The country cannot be added to the custom country watchlist as the list already has too many entities
The date listed end date is before the start date
The date state assigned end date is before the start date
The date status assigned end date is before the start date
The division {0} has no users assigned to it
The division is invalid: {0}
The divisions are invalid:
The EFT type for record {0} is invalid: {1}
The EFT value for record {0} is null or empty
The EFT value length ({0}) for record {1} is greater than the maximum length of {2}
The entity cannot be added to the custom watchlist as the list already has too many entities
The following attachment information is invalid: (followed by one or more of the following statements:) - Notes are required to add attachments - The attachment id must be zero when adding a new attachment - The attachment name is required - The result id is invalid: {0} - The run id is invalid: {0}
The following error occurred while processing your request. Please contact your system administrator.

ServiceFault Error Messages (Continued)

Error Message
The following information is invalid: (followed by one or more of the following statements:) - Invalid Credentials - The DPPA is invalid: {0} - The GLB is invalid: {0} - The list id is invalid: {0} - The list information object is null - The array of list record ids is null or empty - The array of list records is null or empty - The object key is null - The record state object is null - The result id is invalid: {0} - The result information object is null - The array of result record ids is null or empty - The search configuration object is null - The search input object is null - The array of input records is null or empty - The position of the first record is invalid: {0} - The list description is required - The list id must be zero when adding a new list - The list name is required - Search configuration name or custom information is required - The number of record ids ({0}) exceeds the limit of {1} - The number of result ids ({0}) exceeds the limit of {1} - The number of input records ({0}) exceeds the limit of {1} - The number of records to return ({0}) exceeds the limit of {1}
The following list country information is invalid: (followed by one or more of the following statements:) - The aka id(s) must be zero when adding a new country to a list - The location id(s) must be zero when adding a new country to a list - The list id is invalid: {0} - The custom country watchlist record id is invalid: {0} - The custom country watchlist record with the id {0} does not exist - The list to which the record belongs may not be changed - Cannot add a custom country watchlist record to a list of type {0} - The country id must be zero when adding a new country to a list

ServiceFault Error Messages (Continued)

Error Message
The following list entity information is invalid: (followed by one or more of the following statements:) • The additional information id(s) must be zero when adding a new entity to a list • The address id(s) must be zero when adding a new entity to a list • The aka id(s) must be zero when adding a new entity to a list • The identification id(s) must be zero when adding a new entity to a list • The entity type must be one of the following: Business, Individual or Unknown • The entity type must be one of the following: Business, Individual or Vessel • The list id is invalid: {0} • The accept list record id is invalid: {0} • The custom watchlist record id is invalid: {0} • The phone id(s) must be zero when adding a new entity to a list • The accept list record with the id {0} does not exist • The list to which the record belongs may not be changed • Cannot add an accept list record to a list of type {0} • Cannot add a custom watchlist record to a list of type {0} • The entity id must be zero when adding a new entity to a list • The custom watchlist record with the id {0} does not exist
The following list information is invalid: (followed by one or more of the following statements:) • The list id is invalid: {0} • The unique id of the list must not be empty • The unique id of the list may not be changed • The list name is required • The list description is required • The list type is not valid: {0} • The list type may not be changed
The following may not be specified when the result was not written to the database: (followed by one or more of the following statements:) • NewAlertState • NewMatchStates • NewStatus
The FraudPoint® end score is outside the range 0, 300 - 999
The FraudPoint® score end score is before the start score
The FraudPoint® start and end scores are outside the range 0, 300 - 999
The FraudPoint® start score is outside the range 0, 300 - 999
The input records cannot be a mix of EFT, entity and/or ID records
The InstantID® business index end index is before the start index
The InstantID® individual index end index is before the start index
The InstantID® International individual index end index is before the start index
The list id is invalid: {0}
The list information object is null

ServiceFault Error Messages (Continued)

Error Message
The list record criteria object is invalid for a country list search
The list record criteria object is invalid for a custom watchlist search
The list record criteria object is invalid for an accept list search
The list record object is of an unknown type
The list type is not valid: {0}
The list with the primary key {0} does not exist
The MergeDuplicates method only allowed with accept lists
The number of records being added ({0}) exceeds the limit of {1}
The number of result ids ({0}) exceeds the limit of {1}
The number of run ids ({0}) exceeds the limit of {1}
The predefined search {0} contains the provider {1} which can not be used in a Web Services search
The predefined search {0} does not support efts
There are no roles defined
The result cannot be added because it was created from the ACH Gateway Operator and/or the ACH Secondary Screening Indicator
The result cannot be added to the accept list as the list already has too many entities
The result information for result id {0} was not found
The result record {0} was not found
The result record cannot be added. It was searched using the XG - Unstructured format so there is no supporting identifier.
The results end date is before the start date
The role {0} does not contain any valid users
The role {0} does not have access to process results
The role {0} was not found
The role {0} was not found
The run id is invalid: {0}
The runs end date is before the start date
The search configuration {0} was not found
The search criteria for list {0} is not valid
The status {0} is unknown
The total length of EFT values for all records is greater than the maximum length of {0}
The type of the record is not valid: {0}
The user {0} cannot assign results to division {1}
The user {0} has been disabled
The user {0} is not part of the record's division
The user {0} was not found
The value specified for {0} is not a valid date: {1}
The Watchlist end score is outside the range 50 - 100

ServiceFault Error Messages (Continued)

Error Message
The Watchlist score end score is before the start score
The watchlists have not been updated recently
The Watchlist start and end scores are outside the range 50 - 100
The Watchlist start score is outside the range 50 - 100
The Windows SSPI authentication type (double hop) is not supported
This version of the search core is no longer supported
Timed out trying to open: {0}
To open a record you must provide a user or role to assign it to
Type is required in InputAdditionalInfo tag
Type is required in InputAddress tag
Type is required in InputID tag
Type is required in InputPhone tag
Unable to add entity to the accept list because the name or a unique identifier was not specified
User {0} does not have access to add a record to a custom watchlist
User {0} does not have access to add a record to an accept list
User {0} does not have access to update result records
User {0} does not have access to use custom predefined searches
Watchlist capabilities are not enabled for this account
Watchlist files are required to do the Watchlist search
Watchlist matches are required to add the result record to the accept list

ResultError Class

The ResultError class returns any errors that were generated during a search.

Class Properties

Name	Type	Description
Code	Unsigned Integer	This property contains the error code. Not all errors contain an error code.
Message	String	This property contains the error message.

Related Information

[Error Messages](#) on page 188

[Error Codes](#) on page 189

Error Messages

If an error occurs, XG web services may return an error message.



These messages are specific to XG web services. Other messages that are related to XG may be returned as well.

Error Messages

Error Message
Accept List <accept list name> was not found
AppSettings.LoggingPath is missing or empty from the web.config
BI System: Change Password Required
BI System: Failed to login
BI System: Invalid Credentials
BI System: Missing Customer ID
BI System: Missing Password
BI System: Missing User ID
BI System: Password Expiration
Business name cannot be include for entity type Individual
Customer does not have access to Direct Web Services
Customer does not have access to Integrated Web Services
DPPA choice is not valid
Failed to create logging directory '{0}'
Failed to get the accept lists
GLB choice is not valid
Name object can only contain business name for entity type Business
Name object can only contain business name or parsed names for entity type Individual
Name object can only contain full or parsed names for entity type Individual
Name object is null or empty
Name tag does not meet minimum requirements to perform the search Web Services Custom
Name tag is required
Password Change Failed
Predefined search <predefined search> was not found
Predefined search <predefined search name> contains provider '{1}' which can not conduct a search in Web Services
Predefined search <predefined search name> does not support wire.
Predefined search does not contain any searches
Predefined search is not specified
Provider selected can only accept parsed names as valid input
Record {0} was not found
Record Status <record status> is unknown
Request is empty

Error Messages (Continued)

Error Message
Result ID must be greater than 0
Session tag is required for this search to include DPPA and GLB
String <string> was not recognized as a valid DOB
The maximum number of records to process in a search is 100
To open a record you must provide a user or role to assign it to
Unknown Request
User <user name> does not have access to add a record to an accept list
User <user name> does not have access to update this record
WatchList: File <file name> in the array was not found among available files
WatchList: Must specify watchlist files
Watchlist capabilities are not enabled for this account
Web Services encountered an error ({0})
Web Services Predefined Search
Web Services validation error. \'{0}\' type can not be \'None\'

Error Codes

If an error occurs, XG web services may return an error code.



These error codes are specific to XG web services. Other codes that are related to XG may be returned as well.

Bridger Insight XG System Error Codes and Bridger Insight XG Error Codes

Dec Code	Hex Code	Label
0	0x00000000	BRIDGER_INSIGHT_NO_ERROR
1	0x00000001	BRIDGER_INSIGHT_ACCESS_DENIED
2	0x00000002	BRIDGER_INSIGHT_ALREADY_OPEN
3	0x00000003	BRIDGER_INSIGHT_CANNOT_GROW
4	0x00000004	BRIDGER_INSIGHT_FILE_NOT_FOUND
5	0x00000005	BRIDGER_INSIGHT_INVALID_FILE
6	0x00000006	BRIDGER_INSIGHT_INVALID_OBJECT
7	0x00000007	BRIDGER_INSIGHT_INVALID_OPERATION
8	0x00000008	BRIDGER_INSIGHT_INVALID_PARAMETER
9	0x00000009	BRIDGER_INSIGHT_IO_ERROR
10	0x0000000A	BRIDGER_INSIGHT_LOAD_ERROR
11	0x0000000B	BRIDGER_INSIGHT_NO_SERVER_AVAILABLE
12	0x0000000C	BRIDGER_INSIGHT_NO_THREAD
13	0x0000000D	BRIDGER_INSIGHT_NOT_FOUND
14	0x0000000E	BRIDGER_INSIGHT_NOT_OPEN

Bridger Insight XG System Error Codes and Bridger Insight XG Error Codes (Continued)

Dec Code	Hex Code	Label
15	0x0000000F	BRIDGER_INSIGHT_NOT_SUPPORTED
16	0x00000010	BRIDGER_INSIGHT_OUT_OF_MEMORY
17	0x00000011	BRIDGER_INSIGHT_SERIALIZATION_ERROR
18	0x00000012	BRIDGER_INSIGHT_SERVER_CRASHED
19	0x00000013	BRIDGER_INSIGHT_THREAD_ABORTED
20	0x00000014	BRIDGER_INSIGHT_THREAD_ERROR
21	0x00000015	BRIDGER_INSIGHT_THREAD_EXISTS
22	0x00000016	BRIDGER_INSIGHT_TIMED_OUT
23	0x00000017	BRIDGER_INSIGHT_UNHANDLED_ERROR
24	0x00000018	BRIDGER_INSIGHT_UNKNOWN_PATH
25	0x00000019	BRIDGER_INSIGHT_WEB_ERROR
26	0x0000001A	BRIDGER_INSIGHT_XML_ERROR
536870913	0x20000001	BRIDGER_INSIGHT_ALREADY_CONNECTED
536870914	0x20000002	BRIDGER_INSIGHT_CRYPTO_FAILED
536870915	0x20000003	BRIDGER_INSIGHT_DATA_FILE_ERRORS
536870916	0x20000004	BRIDGER_INSIGHT_DATA_TRUNCATED
536870917	0x20000005	BRIDGER_INSIGHT_INVALID_CREDENTIALS
536870918	0x20000006	BRIDGER_INSIGHT_INVALID_DATA
536870919	0x20000007	BRIDGER_INSIGHT_INVALID_FORMAT
536870920	0x20000008	BRIDGER_INSIGHT_INVALID_VALUE
536870921	0x20000009	BRIDGER_INSIGHT_MISMATCHED_DLLS
536870922	0x2000000A	BRIDGER_INSIGHT_PARSE_FAILED
536870923	0x2000000B	BRIDGER_INSIGHT_PASSWORD_EXPIRED
536870924	0x2000000C	BRIDGER_INSIGHT_PASSWORD_CHANGE_FAILED
536870925	0x2000000D	BRIDGER_INSIGHT_PASSWORD_MUST_CHANGE
536870926	0x2000000E	BRIDGER_INSIGHT_PROVIDER_ERROR
536870927	0x2000000F	BRIDGER_INSIGHT_RECORD_LOCKED
536870928	0x20000010	BRIDGER_INSIGHT_TOO_LONG
536870929	0x20000011	BRIDGER_INSIGHT_USER_LOCKED
536870930	0x20000012	BRIDGER_INSIGHT_VALIDATION_FAILED
536870931	0x20000013	BRIDGER_INSIGHT_BATCH_TOO_LARGE
536870932	0x20000014	BRIDGER_INSIGHT_TOO_MANY_ERRORS
536870933	0x20000015	BRIDGER_INSIGHT_INVALID_ENCODING
536870934	0x20000016	BRIDGER_INSIGHT_APPLICATION_CLOSED
536870935	0x20000017	BRIDGER_INSIGHT_INVALID_IP_ADDRESS
536870936	0x20000018	BRIDGER_INSIGHT_INITIALIZATION_FAILED
536870937	0x20000019	BRIDGER_INSIGHT_INVALID_VERSION
536870939	0x2000001B	BRIDGER_INSIGHT_UNMANAGED_OP_FAILED
536870940	0x2000001C	BRIDGER_INSIGHT_INVALID_CLIENT_ID
536870941	0x2000001D	BRIDGER_INSIGHT_INVALID_PRODUCT_CODE

Bridger Insight XG System Error Codes and Bridger Insight XG Error Codes (Continued)

Dec Code	Hex Code	Label
536870942	0x2000001E	BRIDGER_INSIGHT_TRIAL_LAPSED
536870943	0x2000001F	BRIDGER_INSIGHT_USER_DISABLED
536870944	0x20000020	BRIDGER_INSIGHT_RECOVERY_FAILED
536870945	0x20000021	BRIDGER_INSIGHT_INVALID_USER_ID
536870946	0x20000022	BRIDGER_INSIGHT_USER_ID_CHANGE_FAILED
536870947	0x20000023	BRIDGER_INSIGHT_INVALID_PASSWORD
536870948	0x20000024	BRIDGER_INSIGHT_INVALID_PREDEFINED_SEARCH
536870950	0x20000026	BRIDGER_INSIGHT_SEARCH_CORE_ERROR
256	0x00000100	WS_INPUT_ERROR
257	0x00000101	WS_RESULT_PARSE_ERROR
258	0x00000102	WS_TOO_MANY_RECORDS
259	0x00000103	WS_INVALID_PROVIDER
260	0x00000104	WS_SERVICE_NOT_ALLOWED
261	0x00000105	WS_SEARCH_ERROR
262	0x00000106	WS_PREDEFINED_SEARCH_ERROR
263	0x00000107	WS_RESULT_ASSIGNMENT_ERROR

Appendix A

C# Code Examples

LexisNexis[®] provides C# code examples that were created in Microsoft Visual Studio 2010.

AddRecord Code Example

Code examples can help you understand how to use the methods in LexisNexis[®] Bridger Insight[®] XG (XG) web services.

```
////////////////////////////////////  
// Function: AddRecord  
////////////////////////////////////  
  
//Create a client.  
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ListClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
//Choose the list. Look up via searchLists if necessary.  
long ListID = 5;  
  
//Create an array of records to add.  
XgServices.ListRecord[] ListRecords = new XgServices.ListRecord[1];  
  
//Create a record.  
ListRecords[0] = new XgServices.ListRecord();  
ListRecords[0].Entity = new XgServices.EntityInfo();  
ListRecords[0].Entity.Name = new XgServices.EntityName();  
ListRecords[0].Entity.Name.Full = "NewEntity FullName";  
ListRecords[0].Entity.Name.Full = "NewEntity FullName";  
ListRecords[0].Entity.EntityType = XgServices.ListEntityType.Individual;  
  
XgServices.AddRecordResult[] Results = ListClient.AddRecords(Context,ListID,ListRecords);
```

```
////////////////////////////////////  
// END Function: AddRecord  
////////////////////////////////////
```

Related Information

[AddRecords Method](#) on page 130

AddResultRecord Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: AddResultRecord  
////////////////////////////////////  
  
//Create a client.  
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ListClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
//Set the list to add the list to. Look up the list ID via searchLists if necessary.  
long ListID = 1;  
  
//Set the alert state information to change.  
XgServices.ResultInfo result = new XgServices.ResultInfo();  
  
result.NewAlertState = XgServices.AlertStateType.Closed;  
result.NewStatus = "False Positive";  
result.Note = "Adding To Accept List Via Web Services";  
result.ResultID = 688;  
  
long listRecord = ListClient.AddResultRecord(Context, ListID, result);  
  
////////////////////////////////////
```

```
// END Function: AddResultRecord  
////////////////////////////////////
```

Related Information

[AddResultRecord Method](#) on page 131

DeleteRecords Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: DeleteRecords  
////////////////////////////////////  
  
//Create a client.  
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ListClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
//Set the list. Look up the list ID via searchLists if necessary.  
long ListID = 1;  
  
//Set the record to delete.  
XgServices.ArrayOfLong listRecords = new XgServices.ArrayOfLong();  
  
listRecords.Add(31478);  
  
listRecords = ListClient.DeleteRecords(Context, ListID, listRecords);  
  
////////////////////////////////////  
// END Function: DeleteRecords  
////////////////////////////////////
```

Related Information

[DeleteRecords Method](#) on page 132

UpdateRecord Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////
// Function: UpdateRecord
////////////////////////////////////

//Create a client:
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();

//Set the endpoint for the client. It will have a different value for production and test.
ListClient.Endpoint.Address = new System.ServiceModel.EndpointAddress
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set the list. Look up the list ID via searchLists if necessary.
long ListID = 1;

//Set the record to update.
XgServices.ArrayOfLong listRecords = new XgServices.ArrayOfLong();
listRecords.Add(31478);

//Get a record to update.
XgServices.ListRecord[] ReturnedRecords = ListClient.GetRecords(Context, ListID, listRecords);

ReturnedRecords[0].Comments = "Comments Updated Via Web Services";

bool Success = ListClient.UpdateRecord(Context, ReturnedRecords[0]);

////////////////////////////////////
// END Function: UpdateRecord
////////////////////////////////////
```

Related Information

[UpdateRecord Method](#) on page 135

GetRecords Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: GetRecords  
////////////////////////////////////  
  
//Create a client.  
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ListClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
//Set the list. Look up the list ID via searchLists if necessary.  
long ListID = 1;  
  
//Set the list records to be returned.  
XgServices.ArrayOfLong listRecords = new XgServices.ArrayOfLong();  
listRecords.Add(31478);  
listRecords.Add(31479);  
  
XgServices.ListRecord[] ReturnedRecords = ListClient.GetRecords(Context, ListID, listRecords);  
  
////////////////////////////////////  
// END Function: GetRecords  
////////////////////////////////////
```

Related Information

[GetRecords Method](#) on page 133

SearchListRecords Code Example

Code examples can help you understand how to use the methods in XG web services.

```

////////////////////////////////////
// Function: SearchListRecords
////////////////////////////////////

//Create a client.
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();

//Set the endpoint for the client. It will have a different value for production and test.
ListClient.Endpoint.Address = new System.ServiceModel.EndpointAddress
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set the list. Look up the list ID via searchLists if necessary.
long ListID = 1;

//Set the record criteria.
XgServices.ListRecordCriteria Criteria = new XgServices.ListRecordCriteria();
Criteria.DateListedStart = new XgServices.Date();
Criteria.DateListedStart.Day = 1;
Criteria.DateListedStart.Month = 1;
Criteria.DateListedStart.Year = 2000;

//Requires and end.
Criteria.DateListedEnd = new XgServices.Date();
Criteria.DateListedEnd.Day = 1;
Criteria.DateListedEnd.Month = 1;
Criteria.DateListedEnd.Year = 2014;

int StartPosition = 0;
int ReturnCount = 100; //Maximum of 100

XgServices.SearchRecordResults ReturnedRecords = ListClient.SearchListRecords(Context,
ListID, Criteria, StartPosition, ReturnCount);

////////////////////////////////////
// END Function: SearchListRecords
////////////////////////////////////

```

Related Information

[SearchListRecords Method](#) on page 134

UpdateList Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: UpdateList  
////////////////////////////////////  
  
//Create a client.  
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ListClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
//Set the list to get for an update.  
long listID = 1;  
XgServices.ListInfo ListReturn = ListClient.GetList(Context, listID);  
  
//Change list information.  
//Note - Any list information properties that are not included will be deleted from the  
//      existing list settings  
ListReturn.Name = "List Display Name";  
ListReturn.Type = "Accept or Custom";  
ListReturn.Description = "We are updating the description via web services";  
ListReturn.UniqueID = "DB109D85-F167-4E22-918C-760B39AB0CB8";  
  
//Write it back to XG.  
bool Success = ListClient.UpdateList(Context, ListReturn);  
  
////////////////////////////////////  
// END Function: UpdateList  
////////////////////////////////////
```

Related Information

[UpdateList Method](#) on page 126

SearchLists Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: SearchLists  
////////////////////////////////////  
  
//Create a client.  
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ListClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
XgServices.ListCriteria Criteria = new XgServices.ListCriteria();  
  
Criteria.Type = XgServices.ListType.Entity;  
  
XgServices.ListInfo[] ReturnedLists = ListClient.SearchLists(Context, Criteria);  
  
////////////////////////////////////  
// END Function: SearchLists  
////////////////////////////////////
```

Related Information

[SearchLists Method](#) on page 126

AddList Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: AddList  
////////////////////////////////////  
  
//Create a client.  
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ListClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
//Set the list information.  
XgServices.ListInfo NewList = new XgServices.ListInfo();  
NewList.Name = "NewListName";  
NewList.Description = "A New List";  
NewList.Comments = "Comments Go Here";  
NewList.AutoIndex = true;  
NewList.Type = XgServices.ListType.Entity;  
  
long NewListID = ListClient.AddList(Context, NewList);  
  
////////////////////////////////////  
// END Function: AddList  
////////////////////////////////////
```

Related Information

[AddList Method](#) on page 123

IndexList Code Example

Code examples can help you understand how to use the methods in XG web services.

```

////////////////////////////////////
// Function: IndexList
////////////////////////////////////

//Create a client.
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();

//Set the endpoint for the client. It will have a different value for production and test.
ListClient.Endpoint.Address =
new System.ServiceModel.EndpointAddress
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set the list. Look up the list ID via searchLists if necessary.
long listID = 1;
bool Success = ListClient.IndexList(Context, listID);

////////////////////////////////////
// END Function: IndexList
////////////////////////////////////

```

Related Information

[IndexList Method](#) on page 125

DeleteList Code Example

Code examples can help you understand how to use the methods in XG web services.

```

////////////////////////////////////
// Function: DeleteList
////////////////////////////////////

```

```
//Create a client.
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();

//Set the endpoint for the client. It will have a different value for production and test.
ListClient.Endpoint.Address =
    new System.ServiceModel.EndpointAddress
        ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set the list. Look up the list ID via searchLists if necessary.
long listID = 2;

bool Success = ListClient.DeleteList(Context, listID);

////////////////////////////////////
// END Function: DeleteList
////////////////////////////////////
```

Related Information

[DeleteList Method](#) on page 123

GetList Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////
// Function: GetList
////////////////////////////////////

//Create a client.
XgServices.ListMaintenanceClient ListClient = new XgServices.ListMaintenanceClient();

//Set the endpoint for the client. It will have a different value for production and test.
ListClient.Endpoint.Address =
    new System.ServiceModel.EndpointAddress
        ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/ListMaintenance");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
```

```
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set the list. Look up the list ID via searchLists if necessary.
long listID = 1;

XgServices.ListInfo ListReturn = ListClient.GetList(Context, listID);

////////////////////////////////////
// END Function: GetList
////////////////////////////////////
```

Related Information

[GetList Method](#) on page 124

ChangePassword Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////
// Function: ChangePassword
////////////////////////////////////

//Create a client.
XgServices.AccountClient AccountClient = new XgServices.AccountClient();

//Set the endpoint for the client. It will have a different value for production and test.
AccountClient.Endpoint.Address =
    new System.ServiceModel.EndpointAddress
        ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Account");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

bool Success = AccountClient.ChangePassword(Context, "NewValidPassword1!");

////////////////////////////////////
```

```
// END Function: ChangePassword  
////////////////////////////////////
```

Related Information

[ChangePassword Method](#) on page 19

GetDaysUntilPasswordExpires Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: GetDaysUntilPasswordExpires  
////////////////////////////////////  
  
//Create a client.  
XgServices.AccountClient AccountClient = new XgServices.AccountClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
AccountClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Account");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
int RespDays = AccountClient.GetDaysUntilPasswordExpires(Context);  
  
////////////////////////////////////  
// END Function: GetDaysUntilPasswordExpires  
////////////////////////////////////
```

Related Information

[GetDaysUntilPasswordExpires Method](#) on page 20

Search Code Example

Code examples can help you understand how to use the methods in XG web services.

```

////////////////////////////////////
// Function: Search
////////////////////////////////////

//Create a client.
XgServices.SearchClient SearchClient = new XgServices.SearchClient();

//Set the endpoint for the client. It will have a different value for production and test.
SearchClient.Endpoint.Address = new System.ServiceModel.EndpointAddress
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Search");

//Set up context, configuration, and input information.
XgServices.SearchInput Input = new XgServices.SearchInput();

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";
Context.ClientReference = "GUID here Please";
Context.GLB = 5;
Context.DPPA = XgServices.DPPAChoiceType.Choice3;

//Set up the configuration.
XgServices.SearchConfiguration Config = new XgServices.SearchConfiguration();
Config.PredefinedSearchName = "List Screening";

//Set the alert assignment.
Config.AssignResultTo = new XgServices.AssignmentInfo();
Config.AssignResultTo.RolesOrUsers = new XgServices.ArrayOfString();
Config.AssignResultTo.Type = XgServices.AssignmentType.Role;
Config.AssignResultTo.RolesOrUsers.Add("Administrator");
Config.WriteResultsToDatabase = true;

//Set up the input information.
Input.Records = new XgServices.InputRecord[1];
Input.Records[0] = new XgServices.InputRecord();
Input.Records[0].Entity = new XgServices.InputEntity();
Input.Records[0].Entity.EntityType = XgServices.InputEntityType.Individual;
Input.Records[0].Entity.Name = new XgServices.InputName();
Input.Records[0].Entity.Name.First = "Charles";
Input.Records[0].Entity.Name.Last = "Taylor";

```

```
//Provide the input information.
Input.Records[0].Entity.IDs = new XgServices.InputID[1];
Input.Records[0].Entity.IDs[0] = new XgServices.InputID();
Input.Records[0].Entity.IDs[0].Type = XgServices.IDType.SSN;
Input.Records[0].Entity.IDs[0].Number = "1234";
Input.Records[0].Entity.AdditionalInfo = new XgServices.InputAdditionalInfo[1];
Input.Records[0].Entity.AdditionalInfo[0] = new XgServices.InputAdditionalInfo();
Input.Records[0].Entity.AdditionalInfo[0].Type = XgServices.AdditionalInfoType.Incident;
Input.Records[0].Entity.AdditionalInfo[0].Value = "Rejected";

//Call the search.
XgServices.SearchResults Result = SearchClient.Search(Context, Config, Input);

////////////////////////////////////
// END Function: Search
////////////////////////////////////
```

Related Information

[Search Method](#) on page 28

GetDataFileList Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////
// Function: GetDataFileList
////////////////////////////////////

//May want more depending on search parameters.

//Create a client.
XgServices.SearchClient SearchClient = new XgServices.SearchClient();

//Set the endpoint for the client. It will have a different value for production and test.
SearchClient.Endpoint.Address = new System.ServiceModel.EndpointAddress
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Search");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

XgServices.DataFileInfo[] Resp = SearchClient.GetDataFileList(Context);
```

```
////////////////////////////////////  
// END Function: GetDataFileList  
////////////////////////////////////
```

Related Information

[GetDataFileList Method](#) on page 26

GetRunInfo Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: GetRunInfo  
////////////////////////////////////  
  
//Create a client.  
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
long runID =1;  
  
XgServices.RunInfo Response = ResultsClient.GetRunInfo(Context, runID);  
  
////////////////////////////////////  
// END Function: GetRunInfo  
////////////////////////////////////
```

Related Information

[GetRunInfo Method](#) on page 93

SearchRuns Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: SearchRuns  
////////////////////////////////////  
  
//Create a client.  
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
//Set run criteria.  
XgServices.RunsCriteria Criteria = new XgServices.RunsCriteria();  
  
//NOTE: Limit your criteria carefully. A large amount of data is returned for each run.  
Criteria.DateStart = new XgServices.Date();  
Criteria.DateStart.Day = 1;  
Criteria.DateStart.Month = 1;  
Criteria.DateStart.Year = 2013;  
Criteria.SubmitType = XgServices.RunsCriteriaSubmitType.Batch;  
  
XgServices.RunInfo[] Response = ResultsClient.SearchRuns(Context, Criteria);  
  
////////////////////////////////////  
// END Function: SearchRuns  
////////////////////////////////////
```

Related Information

[SearchRuns Method](#) on page 94

DeleteRuns Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: DeleteRuns  
////////////////////////////////////  
  
//Create a client.  
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
//Set the array of runs to delete.  
XgServices.ArrayOfLong runIDs = new XgServices.ArrayOfLong();  
  
//Add at least one run to delete.  
runIDs.Add(689);  
runIDs = ResultsClient.DeleteRuns(Context, runIDs);  
  
////////////////////////////////////  
// END Function: DeleteRuns  
////////////////////////////////////
```

Related Information

[DeleteRuns Method](#) on page 92

DeleteResultRecords Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: DeleteResultRecords
```

```
////////////////////////////////////  
  
//Create a client.  
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");  
  
//Set up the context.  
XgServices.ClientContext Context = new XgServices.ClientContext();  
Context.ClientID = "ClientID";  
Context.UserID = "UserID";  
Context.Password = "Password";  
  
//Set the array of results to delete.  
XgServices.ArrayOfLong resultIDs = new XgServices.ArrayOfLong();  
  
//Add at least one result to delete.  
resultIDs.Add(689);  
  
resultIDs = ResultsClient.DeleteResultRecords(Context, resultIDs);  
  
////////////////////////////////////  
// END Function: DeleteResultRecords  
////////////////////////////////////
```

Related Information

[DeleteResultRecords Method](#) on page 99

GetResultRecords Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////  
// Function: GetResultRecords  
////////////////////////////////////  
  
//Create a client.  
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();  
  
//Set the endpoint for the client. It will have a different value for production and test.  
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress  
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");
```

```
//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set the array of results to return.
XgServices.ArrayOfLong resultIDs = new XgServices.ArrayOfLong();

//Add at least one result.
resultIDs.Add(514156);

XgServices.ResultRecord[] Results = ResultsClient.GetResultRecords(Context, resultIDs);

////////////////////////////////////
// END Function: GetResultRecords
////////////////////////////////////
```

Related Information

[GetResultRecords Method](#) on page 100

SetRecordState Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////
// Function: SetRecordState
////////////////////////////////////

//Create a client.
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();

//Set the endpoint for the client. It will have a different value for production and test.
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set state values.
```

```
XgServices.RecordState State = new XgServices.RecordState();
State.AlertState = XgServices.AlertStateType.Closed;
State.Status = "Application Declined outside of Compliance";
State.Note = "Closing from Web Services";

//Set a result to apply the state to.
long resultID = 514623;

bool Success = ResultsClient.SetRecordState(Context, resultID, State);

////////////////////////////////////
// END Function: SetRecordState
////////////////////////////////////
```

Related Information

[SetRecordState Method](#) on page 101

SearchResultRecords Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////
// Function: SearchResultRecords
////////////////////////////////////

//Create a client.
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();

//Set the endpoint for the client. It will have a different value for production and test.
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set query criteria.
XgServices.ResultsCriteria Criteria = new XgServices.ResultsCriteria();

//Set required start date.
Criteria.DateStart = new XgServices.Date();
Criteria.DateStart.Day = 6;
```

```
Criteria.DateStart.Month = 25;
Criteria.DateStart.Year = 2013;

//Set required end date.
Criteria.DateEnd = new XgServices.Date();
Criteria.DateEnd.Day = 6;
Criteria.DateEnd.Month = 27;
Criteria.DateEnd.Year = 2013;

Criteria.ID = new XgServices.EntityCriteriaID();
Criteria.ID.Number = "12345";
Criteria.ID.Type = XgServices.IDType.Account;

Criteria.RecordState = new XgServices.ResultCriteriaRecordState();
Criteria.RecordState.CurrentStatus = "False Positive";

long ReturnStart = 0;
int ReturnCount = 100;

XgServices.SearchRecordResults Results = ResultsClient.SearchResultRecords(Context, Criteria,
    ReturnStart, ReturnCount);

////////////////////////////////////
// END Function: SearchResultRecords
////////////////////////////////////
```

Related Information

[SearchResultRecords Method](#) on page 101

AddAttachment Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////
// Function: AddAttachment
////////////////////////////////////

//Create a client.
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();

//Set the endpoint for the client. It will have a different value for production and test.
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");
```

```
//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

XgServices.Attachment Attachment= new XgServices.Attachment();

Attachment.FileType = XgServices.AttachmentFileType.TXT;
Attachment.Data = Encoding.ASCII.GetBytes("This is a text file. Honestly.");

//Set the result to add the attachment to.
Attachment.ResultID = 514156;

//Set the required run ID. Look up via GetResultRecords if necessary.
Attachment.RunID = 683;

//Provide a name. The extension must match the file type.
Attachment.Name = "NameOfFile.txt";
Attachment.Notes = "Note is required";

long attachmentid = ResultsClient.AddAttachment(Context, Attachment);

////////////////////////////////////
// END Function: AddAttachment
////////////////////////////////////
```

Related Information

[AddAttachment Method](#) on page 119

DeleteAttachment Code Example

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////
// Function: DeleteAttachment
////////////////////////////////////

//Create a client.
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();

//Set the endpoint for the client. It will have a different value for production and test.
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress
("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");
```

```
//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set the result and attachment ID.
long ResultID = 514156;
long attachmentID = 4;

bool Success = ResultsClient.DeleteAttachment(Context, attachmentID, ResultID);

////////////////////////////////////
// END Function: DeleteAttachment
////////////////////////////////////
```

Related Information

[DeleteAttachment Method](#) on page 119

GetAttachment Code Sample

Code examples can help you understand how to use the methods in XG web services.

```
////////////////////////////////////
// Function: GetAttachment
////////////////////////////////////

//Create a client.
XgServices.ResultsClient ResultsClient = new XgServices.ResultsClient();

//Set the endpoint for the client. It will have a different value for production and test.
ResultsClient.Endpoint.Address = new System.ServiceModel.EndpointAddress
    ("https://bridgerinsight.lexisnexis.com/LN.WebServices/11.3/xgservices.svc/Results");

//Set up the context.
XgServices.ClientContext Context = new XgServices.ClientContext();
Context.ClientID = "ClientID";
Context.UserID = "UserID";
Context.Password = "Password";

//Set the result and attachment ID.
long ResultID = 514156;
long attachmentID = 4;
```

```
XgServices.Attachment Attachment = ResultsClient.GetAttachment(Context, attachmentID,  
    ResultID);
```

```
////////////////////////////////////  
// END Function: GetAttachment  
////////////////////////////////////
```

Related Information

[GetAttachment Method](#) on page 120

Index

A

AddAttachment method [119](#)
AddList method [123](#)
AddRecordResult class [138](#)
AddRecords method [130](#)
AddResultRecord class [131](#)
AssignmentInfo class [31](#)
Attachment class [121](#)
AuditItem class [60](#)

C

ChangePassword method [19](#)
ClientContext class [21](#)
CountryAKA class [139](#)
CountryCriteria class [139](#)
CountryInfo class [140](#)
CountryLocation class [140](#)

D

DataFileInfo class [27](#)
Date class [95](#), [105](#), [140](#), [168](#), [173](#)
DeleteAttachment method [119](#)
DeleteList method [123](#)
DeleteRecords method [132](#)
DeleteResultRecords method [99](#)
DeleteRuns method [92](#)
DOB match level [79](#)

E

EntityAdditionalInfo class [174](#)
EntityAddress class [174](#)
EntityAKA class [175](#)
EntityCriteria class [141](#)
EntityCriteriaAdditionalInfo class [178](#)
EntityCriteriaAddress class [178](#)
EntityCriteriaID class [179](#)
EntityID class [176](#)

EntityInfo class [142](#)
EntityMatchInfo class [143](#)
EntityName class [176](#)
EntityPhone class [177](#)

F

FPAAddress class [68](#)
FPInput class [68](#)
FPRedFlag class [68](#)
FPResultCriteriaRedFlag class [105](#)
FPRiskIndicator class [69](#)
FraudPointConfiguration class [31](#)
FraudPointResultCriteria class [106](#)
FraudPointResults class [69](#)

G

GetAttachment method [120](#)
GetDataFileList method [26](#)
GetDaysUntilPasswordExpires method [20](#)
GetList method [124](#)
GetRecords method [133](#)
GetResultRecords method [100](#)
GetRunInfo method [93](#)

I

IIDAddress class [71](#)
IIDAddressHistory class [71](#)
IIDAlternateName class [71](#)
IIDFEINMatch class [72](#)
IIDFiling class [72](#)
IIDInput class [73](#)
IIDIResultCriteriaIndividual class [107](#)
IIDRedFlag class [73](#)
IIDResultCriteriaBusiness class [108](#)
IIDResultCriteriaIndividual class [109](#)
IIDResultCriteriaRedFlag class [110](#)
IIDReversePhone class [74](#)
IIDRiskIndicator class [74](#)

[IIDSSNInfo class 74](#)
[IIDVerificationIndicators class 75](#)
[IIDWatchlist class 75](#)
[IndexList method 125](#)
[InputAdditionalInfo class 169](#)
[InputAddress class 169](#)
[InputBasicID class 52](#)
[InputEFT class 52](#)
[InputEntity class 53](#)
[InputID class 170](#)
[InputName class 172](#)
[InputPhone class 172](#)
[InputRecord class 54](#)
[InputText class 55](#)
[InstantIDBusinessResults class 76](#)
[InstantIDConfiguration class 33](#)
[InstantIDIndividualResults class 77](#)
[InstantIDIntlResultCriteria class 111](#)
[InstantIDResultCriteria class 111](#)

L

[ListCriteria class 127](#)
[ListInfo class 128](#)
[ListRecord class 144](#)
[ListRecordCriteria class 145](#)

O

[OFACInstitutionInfo class 79](#)
[OFACReportInfo class 80](#)

P

[ParsedACH class 148](#)
[ParsedACHAdditionalInfo class 149](#)
[ParsedACHAddress class 149](#)
[ParsedACHField class 150](#)
[ParsedACHID class 150](#)
[ParsedACHNote class 150](#)
[ParsedACHParty class 151](#)
[ParsedFedwire class 154](#)
[ParsedFedwireAdditionalInfo class 154](#)
[ParsedFedwireAddress class 154](#)
[ParsedFedwireAmount class 155](#)

[ParsedFedwireField class 156](#)
[ParsedFedwireID class 156](#)
[ParsedFedwireNote class 156](#)
[ParsedFedwireParty class 157](#)
[ParsedFedwirePhone class 158](#)
[ParsedISO20022 class 158](#)
[ParsedISO20022AdditionalInfo class 159](#)
[ParsedISO20022Address class 159](#)
[ParsedISO20022Field class 160](#)
[ParsedISO20022ID class 160](#)
[ParsedISO20022Note class 161](#)
[ParsedISO20022Party class 161](#)
[ParsedISO20022Phone class 162](#)
[ParsedSWIFT class 163](#)
[ParsedSWIFTAdditionalInfo class 164](#)
[ParsedSWIFTAddress class 164](#)
[ParsedSWIFTAmount class 165](#)
[ParsedSWIFTField class 165](#)
[ParsedSWIFTID class 166](#)
[ParsedSWIFTNote class 166](#)
[ParsedSWIFTOriginalMessage class 166](#)
[ParsedSWIFTParty class 167](#)

R

[RecordState class 61](#)
[ResultCriteriaAccount class 114](#)
[ResultCriteriaEFT class 114](#)
[ResultCriteriaRecordState class 115](#)
[ResultError class 187](#)
[ResultInfo class 146](#)
[ResultRecord class 62](#)
[ResultRecordDetails class 63](#)
[ResultsCriteria class 112](#)
[RunInfo class 96](#)
[RunsCriteria class 95](#)

S

[Search method 28](#)
[SearchConfiguration class 33](#)
[SearchInput class 55](#)
[SearchListRecords method 134](#)
[SearchLists method 126](#)
[SearchRecordResults class 117, 147](#)

SearchResultRecords method [101](#)
SearchResults class [67](#)
SearchRuns method [94](#)
ServiceFault class [181](#)
SetRecordState method [101](#)

T

Time class [117](#)

U

UpdateList method [126](#)
UpdateRecord method [135](#)

W

WatchlistAutomaticFalsePositiveRules class [35](#)
WatchlistConfiguration class [41](#)
WatchlistDataFile class [40](#)
WatchlistEFTMatchOptions class [42](#)
WatchlistGeneralOptions class [44](#)
WatchlistIntelligentMatchDecisionSolution class [46](#)
WatchlistMatchDispositionRules class [47](#)
WatchlistMatchOptions class [49](#)
WatchlistMatchState class [67](#)
WatchlistPaymentScreeningOptions class [50](#)
WatchlistResultCriteria class [117](#)
WatchlistResults class [80](#)
WatchlistSearchCriteria class [50](#)
WatchlistSearchCriteriaGroup class [51](#)
WLAddressMatch class [81](#)
WLCitizenshipMatch class [81](#)
WLCountryDetails class [82](#)
WLDOBMatch class [82](#)
WLEntityDetails class [83](#)
WLIDMatch class [84](#)
WLMatch class [84](#)
WLMatchConflicts class [88](#)
WLMatchFile class [88](#)
WLMatchFprDetails class [89](#)
WLMatchSummary class [89](#)
WLMatchSummaryItem class [90](#)
WLPhoneMatch class [90](#)