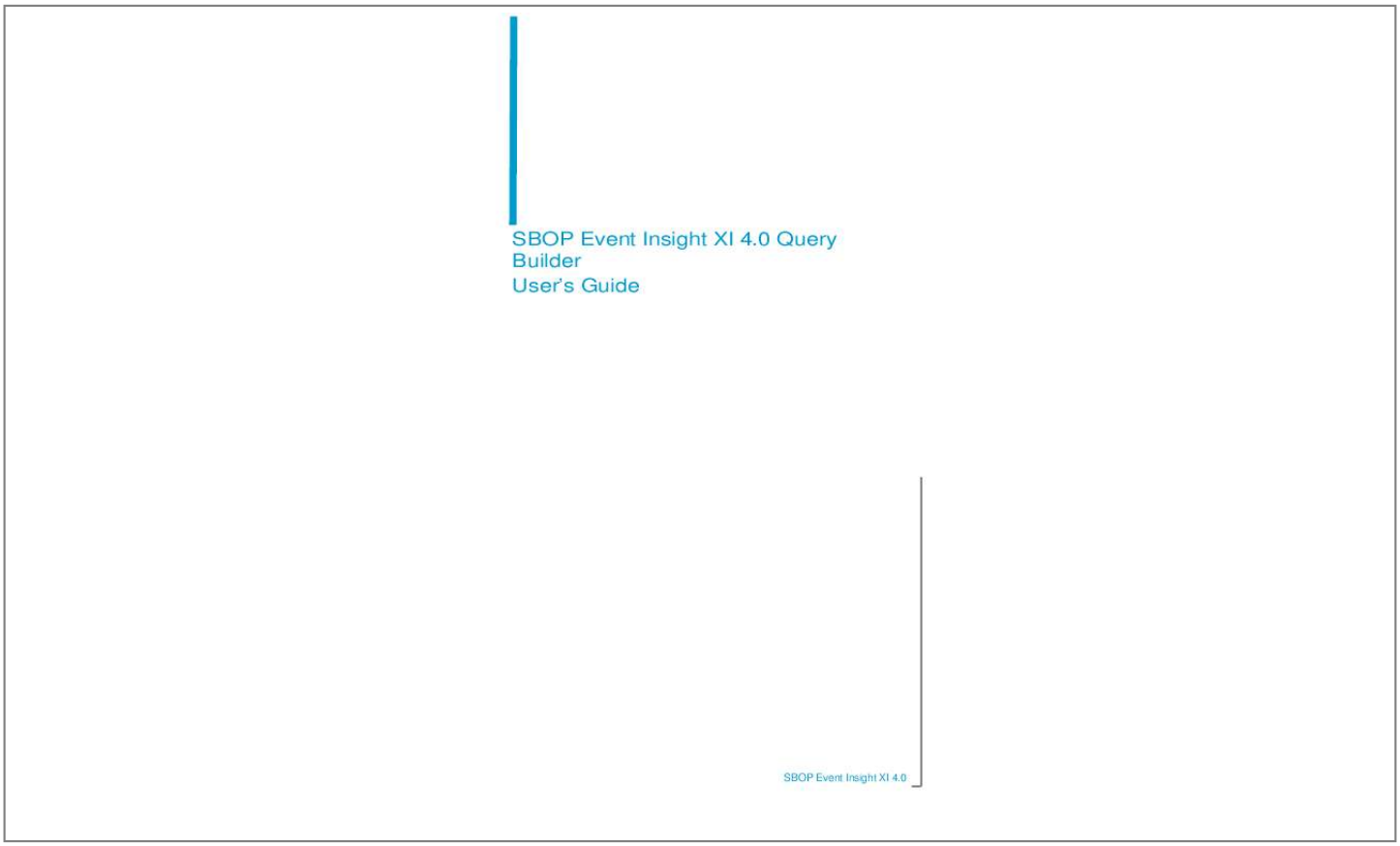

Manual abstract:

Business Objects is an SAP company. @@Data contained in this document serves informational purposes only. National product specifications may vary. These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any Nothing herein should be construed as constituting an additional warranty. 2010-10-01 Contents Chapter 1 Query Builder ....

0 Query Builder User's Guide Query Builder 1 1\ Query Builder Overview Overview Query Builder is a client application used to construct, submit, and track query status. You can subscribe to events and retrieve the results in tabular form. Starting Query Builder To start Query builder, choose Start > Programs > SAP EventInsight > QueryBuilder > QueryBuilder. Understanding the Query Builder user interface The following graphic shows Query Builder with all commonly used views. 6 SBOP Event Insight XI 4.0 Query Builder User's Guide Query Builder Overview The following table describes the Query Builder views that are commonly used. \1 Category General View Console Palette Properties Description Displays log of actions. Displays selection action types, connection tool, and node types. Displays properties of the selected object. Displays error messages that are processed during application runtime (for example, a graph failed to load due to an exception).
Displays graph parameters that are used in graph node attribute placeholders. Graph parameters in query nodes are specified using the ${parameter} format. For example, a password in a DB Output node may be specified as ${passwd}, and in the Graph Parameters view you set it to "mypwd" by specifying the parameter name/value as passwd/mypwd. Displays received events. Displays query graphs stored on the metadata server.
Provides format for notifications to users. Displays graph validation messages. If a graph validations fails (for example, some of the nodes have unconnected ports, or some node parameters are not set correctly) then this view displays errors and the appropriate location (such as the path to a graph or a source). Displays the query graph. LE Common Messages Query Builder Views Graph Parameters Graph Results Graph Tree Notification Templates Validation Messages n/a Graph editor SBOP Event Insight XI 4.
0 Query Builder User's Guide 7 1\ Query Builder Overview Understanding the palette Query graphs are used to describe event processing. A node can be an event source, receiver, sender, or filter, depending on the node type. You add a node to a query graph using the palette. The following table describes the palette buttons for working with query graphs: Button group Selection and connection Button Select Connection Description Activates selection mode, in which you can select a query graph node. Activates node connection mode, in which you can connect output and input port of two different query nodes in a graph. Write direct CQL to correlate events using the CEP engine. Used to send events to a database. Specifies input event types. Filter events based on specific criteria. Simplifies events by outputting a selected subset of the input event Redirects events displayed on different communication devices, such as mobile phone or email.

Marks events to send to UI and displays events in the graph result view. Subscribe to another event within a graph execution. Write java code to filter events. Write java script to filter events Drag and drop CQL to correlate events. Lets you output events to a configured JMS destinations. Graph nodes CQL node DB Output node Event node Filter node Flattening node Notification node Output node Trigger node Java node Java Script node Semantic node JMS node Changing Query Builder Preferences 1. Select Window > Preferences. The Preferences window opens. 2. Modify the parameter values as required.
3. To apply the changes, click Apply. 4. To restore the default settings, click Restore Defaults. 5.
To save the changes, click OK. 8 SBOP Event Insight XI 4.0 Query Builder User's Guide Query Builder Overview \1 Changing the Metadata Server Address To connect Query Builder to a metadata server other than the default servers, change the Query Builder parameters as follows: 1. Select Window > Preferences. The Preferences window opens.
2. Select Query Builder. 3. Set the metadata server address in the host:port format. If the value is incorrect or missing, the auto discovery service is used to find a metadata server on the local network. 4. To save the changes, click OK. Changing the Security Service Address To connect Query Builder to a security service other than the default service, change the Query Builder parameters as follows: 1. Select Window > Preferences. The Preferences window opens.

2. Select Query Builder. 3. Set the security service server address in the host:port format. If the value is incorrect or missing, the auto discovery service is used to find a security service on the local network. 4. @@Select Window > Preferences. The Preferences window opens. 2. Select Query Builder.
3. Set the CEP engine server address in the host, port fields.

4. @@@@Select Window > Preferences. The Preferences window opens.
2. Select Query Builder > Query Manager. 3. @@@@The new graph will be created in the selected group. 2.
Enter the name of the graph. @@Click on Find to select a global schema instance. The Select Global Schema Instance window opens. 4.
@@@@@@@@Enter the new grarameter), the graph is validated. @@@@Doubleclick on validation message opens graph where error is found.
@@@@In the Graph Tree, right-click the graph and select Start Graph. @@@@@@@A trigger node has one input and one output port.
@@@@@@@@This mapping is done with simple expressions. @@Triggered subscription termination is also configurable; it's based on the received event count or timeout.

Creating a trigger node You can create a trigger node, as any other node, in Query Builder. This is the event type for which the subscriptions will be triggered on incoming events. 1. In Query Builder, create a new graph or choose an existing graph. 2. Select Trigger Node from the palette and click the graph editor. The Select Element window opens. 3. Connect the Trigger node's input port with some other node's output port for which the event schema is defined, select the Trigger node and open properties view. The node has the following properties: Property Event type Strategy Description The event type you have chosen when creating a Trigger node; you can always change it later from this view.
How the subscription is managed. There are 2 strategies: event count and timeout. Event count means that subscription will be terminated after the number of received events reaches the value which is specified in Strategy value property. Timeout strategy means that subscription will be terminated after number of milliseconds specified in Strategy value property. A global timeout for a triggered subscription.
No matter what strategy was chosen, the subscription will be terminated after number of milliseconds specified in this property; in most cases the subscription should be terminated by the Strategy + Strategy value combination, thus Time to live value should be set to big enough. Time to live All other properties are the event type parameters which are passed when subscribing to the event type. These properties displayed vary according to the event type you have chosen, it's fully possible that event type has no parameters, in this case no additional properties appear. SBOP Event Insight XI 4.0 Query Builder User's Guide 17 4\ Graph nodes Trigger node There are two ways to set an event type parameter: Enter the value.
With this method, the value is a constant, and is always passed when subscribing disregarding the contents of the incoming event. Use a simple expression which will bind an attribute in the incoming event to the event type parameter. Expression specification is: ->attribName1>attribName2->...->attribNameN where attribName is an attribute name in the incoming event. Note that if an attribute name in the expression is succeeded by -> that means that this attribute must have a nested structure and contain sub attributes. Last attribute name in the expression must correspond to a flat attribute, like Integer, String, etc. If an attribute in the incoming event is a list it's possible to access it's elements by a get() token, there are 3 available parameters to pass to this token: an arbitrary integer, which represents the index in the list, first, which represents the first element in the list and last, which respectively points to the last element in the list. Index out of bounds problem is treated inside the Trigger node and in this case no value will be set to a corresponding event type parameter.

An example of valid expression with get(): ->attribName1->get(3)-->attribName2->get(last). Here, attribName1 and attribName2 are lists. Note that any type of error in the expression will lead to an empty event parameter passed at the subscription time. After all properties have been set, the trigger node will become valid; its output port schema will coincide with the Event type schema. All events produced by the Trigger node will be sent to its output port. 18 SBOP Event Insight XI 4.0 Query Builder User's Guide Graph nodes Flattening node \4 Flattening node A flattening node can process events with complex structures and output events with simpler structures. A flattening node has one input and one output port. On every incoming event (an event on an input port) the flattening node extracts a user specified element and produces output event(s) from this element. The node properties: Property path Description The element from the input event the flattening node must process for output.
The output data type from the flattening node will depend on the data type of the element defined by the path property. If path is set to a list element then for every incoming event each entry in the list element will correspond to an output event. The list is effectively flattened into a sequence of individual events. Using a flattening node On the Query Builder UI, Flattening node is available in the palette. It can be used to turn a list of data in one event into a sequence of events.
Simplify event structures before using CQL node for processing. (CQL node only accepts events with certain data types SBOP Event Insight XI 4.0 Query Builder User's Guide 19 4\ Graph nodes Notification node Notification node Notification node provides Event Insight users with notifications via instant messages, e-mail, and SMS. In a Query Builder, a Query Node is provided for notification management. The table below lists Event Insight components that are relevant to the functionality of Notification feature.
Component Notification node Description Query Node visible in Query Builder palette; provides possibility to configure condition, type, and contents of notifications. A bundle which must be deployed and started on a bundle manager (or any other BundleManager within which context Notification nodes are supposed to be processed); during event flow processing it generates and distributes service-level notification events to Event Insight network. A bundle which provides Notification Service consumer capable receiving and processing servicelevel notification events; it is responsible for physical notifications sending (and thus responsible for intercommunication with third party notification services). NotificationManager NotificationService Using a notification node On the Query Builder UI, Notification node is available in the palette. It can be used as a sink node in a query graph, if the user wants to be notified based on events that arrive on the Notification node. 20 SBOP Event Insight XI 4.0 Query Builder User's Guide Graph nodes Notification node \4 Notification node parameters Parameter E-mail Jabber recipient Phone number Online Offline Other SMS Gateway Template Path Service ID Description Recipient's e-mail address.

Recipient's Jabber ID. Recipient's phone number. Notification used when recipient's Jabber client is online.

Notification used when recipient's Jabber client is offline. Notification used when recipient's Jabber client is in any other status. SMS gateway to be used to send short messages to the recipient (using specified Phone number). Path to template in metadata repository to be used to format notification message. Service ID that will process service-level notification events generated by notification manager. This gives possibility for user to specify on which Bundle Manager(s) his notification messages will be sent. Notification forms You can specify the following forms of notifications: E-mail Jabber SMS None (no notifications sent) Templates With templates, you can format notification messages and limit user notification sending possibilities. Users which submit Template Path, must have Read action on this resource to have notifications sent by Notification Service. Notification Templates view provides exploring and management functionality. Templates and their groups can be created, deleted, and modified in this view.
For each template, a separate editing view will be opened. Templates may contain placeholders, which will be replaced by real values taken from event contents during Event Insight runtime. Placeholder's must be specified in format ${placeholder_name}. SBOP Event Insight XI 4.0 Query Builder User's Guide 21 4\ Graph nodes Notification node SMS carriers You can select SMS carriers (also named gateways) from the predefined list.
22 SBOP Event Insight XI 4.0 Query Builder User's Guide Graph nodes JMS Output Node \4 JMS Output Node JMS Output Node gives possibility to perform same functionality as JMS Output Adapter, but configuring it within a query graph: subscription to Event Insight event type translation of received Event Insight query events to JMS messages publishing of generated JMS Messages to specified JMS topic or queue, prior connecting to it based on source and configuration objects data JMS Ouput node parameters Parameter Attribute Field Name Connect Attempt Timeout Description Attribute name that will be used to access data in query events for data retrieval, which will be used to create JMS messages. Connect attempt timeout in milliseconds; if an adapter fails to connect to JMS Provider within this time frame, connection establishment will fail; the default (and minimum) is 5000 (5 seconds). Connection factory name that will be used to retrieve JMS Connection Factory during JNDI lookup. JMS Destination name, which will be used to publish JMS Messages. JMS Destination type (topic or queue), which will be used to publish JMS Messages. @@@@@@Login for creating connections to secured destinations. Password for creating connections to secured destinations. @@Transformation strategy that will be applied to query events and that determines what kind of JMS Message will be produced on output. Connection Factory Destination Name Destination Type ECF Provider ID Initial Context Factory Login Password Provider URL Transformation Strategy SBOP Event Insight XI 4.0 Query Builder User's Guide 23 4\ Graph nodes JMS Output Node Transformation Strategies The table below summarizes event transform strategies, as well as meaning of attribute name in configuring strategy: Strategy TEXT_MESSAGE STREAM_MESSAGE Attribute Type Meaning Global event schema's attribute must be of type String. Global event schema's attribute must be of type ListType<?>, where <?> can be of any type. Note: Complex types are not supported by specification and most JMS providers, whilst some JMS providers support complex structures consisting of Maps and Lists. BytesMessage ObjectMessage MapMessage Global event schema's attribute must be of type ListType<Byte>. Global event schema's attribute may be of any type.

- (attribute name is not used) 24 SBOP Event Insight XI 4.0 Query Builder User's Guide Integrating with a Complex Event Processing (CEP) engine 4 4\ Integrating with a Complex Event Processing (CEP) engine Editing a CEP query Editing a CEP query 1. Double-click the CQL node. When you make changes in the CEP editor, the related CEP node query is also updated. You can also edit the CEP query in the Properties view. After you select a CEP query, the editor opens. The CEP text editor highlights reserved CEP keywords (for example, select and insert). Here are some guidelines for using the CQL node: The query may contain only one SELECT statement or one CREATE ACTIVE TABLE statement. A SELECT statement defines a continuous query and requires an output port. A CREATE ACTIVE TABLE statement defines an active table, which is a table that will be updated with the data produced by a query following the CREATE ACTIVE TABLE statement.
No output port is required. For each input, an auto generated attribute time_stamp is added. You can view this attribute in the port properties. 2. After you enter or change a query, click Save.
If the query is a valid CEP query and the port schemas are correct, the CQL node is displayed in green. Defining ports You can define ports and port schemas for a CEP node. 1. Right-click the node and select Add Port. The Add Port window opens.
You can define schemas only for output ports. For input ports, the schema is calculated from the connected node port's schema, and you cannot edit the event schema. You can define a port schema by adding attributes one by one or can be created from exiting event type schema. 2. To create a schema from exiting event type schema, click the Create From Existing Schema button. The Select Event Schema window is being displayed. Event types can be either local or global. Select the event type with the appropriate schema. 3. To edit existing port information, use the Properties view, right-click the node and select Edit port, or double-click the port.

This functionality is available only for CEP ports. Limitations CEP parameter binding to default values is not supported because Event Insight has no parameter logic. Simple parameter value replacement would destroy the purpose of parameters (the ability to update them during query execution). Event Insight supports bindings between input and output streams (in any combination). Event Insight does not retain information about bindings that contain local streams as target or sink; these relations cannot be represented using Query Graph. Only a CEP project that uses schema definitions in queries is supported. There is no type inference in the case of multiple queries in one module. 26 SBOP Event Insight XI 4.0 Query Builder User's Guide Index C CEP query, 26 connection adding, 14 CQL node, 26 F Flattening node, 19 N Notification node, 20 T Trigger node, 17 P palette, 8 Q Query Builder overview, 6 using, 12 query graph editing, 14 4 .