



orbis software
THE POWER TO MAKE IT HAPPEN

**TaskCentre® v4.5
SalesLogix Connector Tool
White Paper**

ABOUT SALESLOGIX CONNECTOR TOOL 1

INTRODUCTION 3

 SalesLogix Version Support 3

ARCHITECTURE 4

SALESLOGIX CONNECTOR TOOL..... 5

 Introduction 5

 Global Configuration..... 5

 Step Configuration 5

 Filter Links 9

 Examples..... 9

 Transaction Control10

 Tool Output.....10

 Runtime11

 Errors and failures11

SALESLOGIX CONNECTOR WEB SERVICE 11

 Introduction12

 Schema Creation12

 Data Processing13

 How data is processed for each operation13

 Transactions.....14

 Data Security14

 SalesLogix Security15

 SECCODEID16

 Cascading Deletes.....16

 Synchronisation17

 Attachments17

About SalesLogix Connector Tool

For further information, please contact your TaskCentre distributor.

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein. Accuracy of any information presented after the date of publication is not guaranteed.

This White Paper is for informational purposes only. NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, ARE MADE AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Orbis Software.

Orbis Software may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except where expressly provided in any written license agreement from Orbis Software, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

©2008 Orbis Software. All rights reserved.

TaskCentre is a trademark of Orbis Software Ltd.

SalesLogix is a trademark of Sage (UK) Limited.

Portions of TaskCentre Tools for SalesLogix are ©2008 Codestone Software Ltd. All rights reserved.

All other trademarks are acknowledged as the property of their respective owners.

Introduction

The aim of the TaskCentre SalesLogix Connector Tool project is to create a Tool that gives a simple, straightforward way of providing connectivity between TaskCentre and SalesLogix. Currently, it is possible for TaskCentre to read data from a SalesLogix system via an OLE DB query Tool, using the SalesLogix OLE DB driver. It is not straight forward to write data to SalesLogix via TaskCentre however and large amounts of script are needed to achieve this.

The SalesLogix Connector Tool will use XML as its input and output data format. Interoperation with standard TaskCentre Tools will be achieved by the use of TaskCentre XML tools, which will be released at the same time as the SalesLogix tool.

Handling of events within SalesLogix can be achieved by use of a standard TaskCentre trigger event on the SalesLogix database tables.

The Connector Tool itself will not directly connect to SalesLogix. All data transfer to/from SalesLogix will be done via the SalesLogix OLE DB Provider, and this takes place in a Web Service that will be distributed with the Connector Tool. This allows distribution of the components so that the TaskCentre server does not have to be on the same machine as the SalesLogix OLE DB Provider.

SalesLogix Version Support

A final decision has yet to be made, but at this point it is intended to support SalesLogix versions 6.2, 7.0 and 7.2. SalesLogix backend RDBMS support will initially be SQL 2000, SQL 2005, Oracle 9i R2 and Oracle 10g R2.

SalesLogix 7.5 is now entering beta, and further information about support for this will be following when determined.

Architecture

Figure 1 below provides a high-level system architecture diagram of the SalesLogix connector tool within the context of TaskCentre and SalesLogix. Items in blue are part of the SalesLogix Connector Tool.

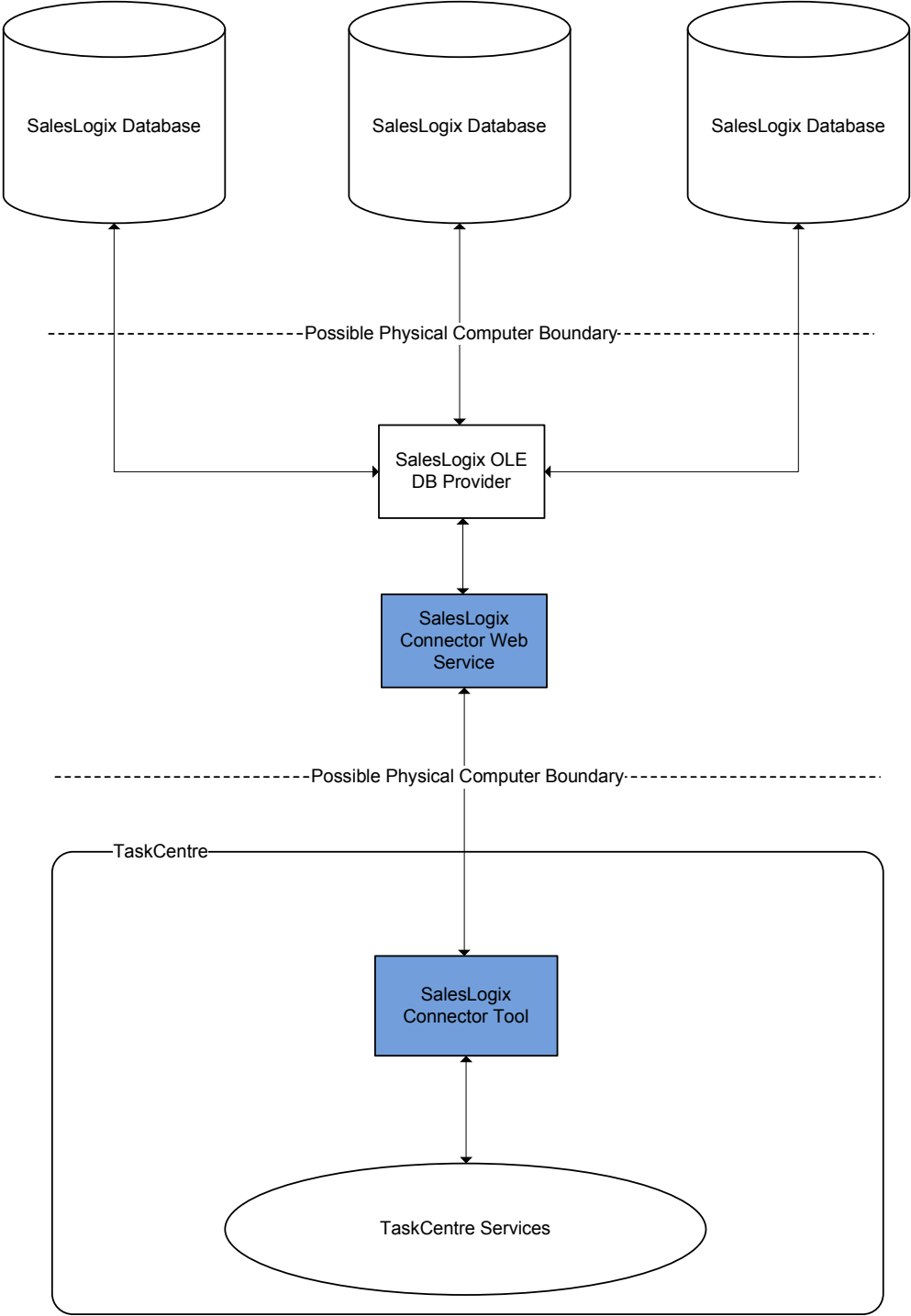


Figure 1 – System Architecture Diagram

SalesLogix Connector Tool

Introduction

The SalesLogix Connector Tool is used to operate with data from/to a SalesLogix database. The operations available are Get, Update, Add and Delete.

The Tool transforms incoming XML data into an XML format for the specific SalesLogix object and operation selected. This XML is then passed to the Connector Web Service which does the actual communication with SalesLogix via the dedicated OLE DB Provider. The response is exposed as XML from the Tool, as well as a special TaskCentre recordset containing information on whether the operation succeeded or failed for the data passed in.

The Tool does not accept a TaskCentre recordset directly as input. A TaskCentre recordset can, however, be processed by the Recordset to XML Tool first to transform it into XML. This is one of the forthcoming TaskCentre XML tools which will be released at the same time as the SalesLogix Connector Tool.

Global Configuration

The global configuration dialogs for the Tool shown in Figure 2 below are where connections are defined for each SalesLogix database that is to be used by TaskCentre. The information for each connection includes SalesLogix Server, database, username, password and so forth. When the connection is set up, the available objects and fields for the selected SalesLogix database are retrieved, and a schema for each object is created. This information is used in the Step configuration. If there is a change to the SalesLogix database (alteration to the user defined fields/tables, for instance) then the object schemas can be refreshed by editing the connection

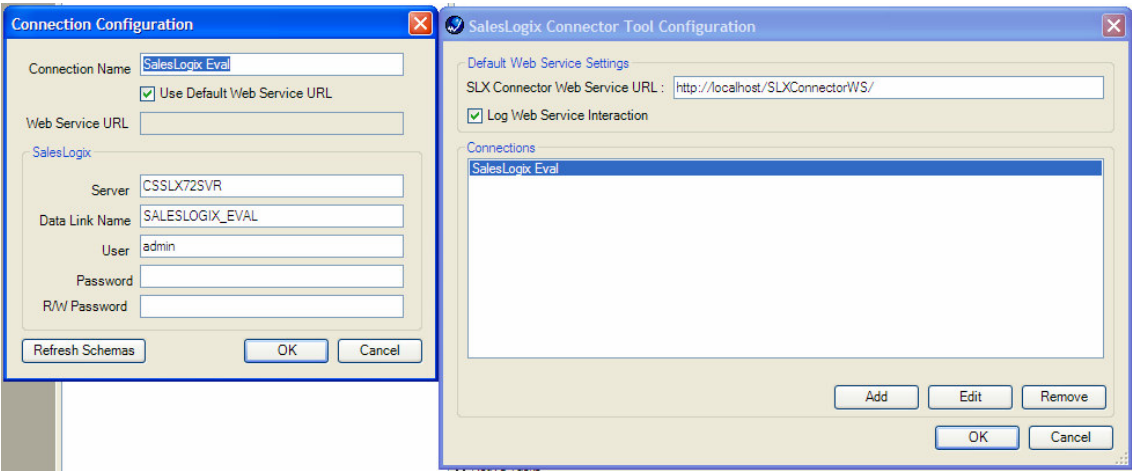


Figure 2 – SalesLogix Connector Tool Configuration and Connection Configuration dialogs

Step Configuration

At configuration time, when the Tool is added to a Task the Step will allow the user to select the schema of the XML data that is to be passed to it at runtime through the General tab of the SalesLogix Connector dialog. See Figure 3 below.

This schema will have come from another Step in the same Task.

A SalesLogix connection is also required for the step. This is selected from a list of connections previously set up through the Global Configuration dialogs for the tool.

For the SalesLogix connection a global connection can be selected or a TaskCentre variable can be dragged in to create a dynamic connection. If the dynamic option is used, then at runtime the text in the variable should match the name of one of the SalesLogix connections setup in the Global Configuration, and this connection is then used.

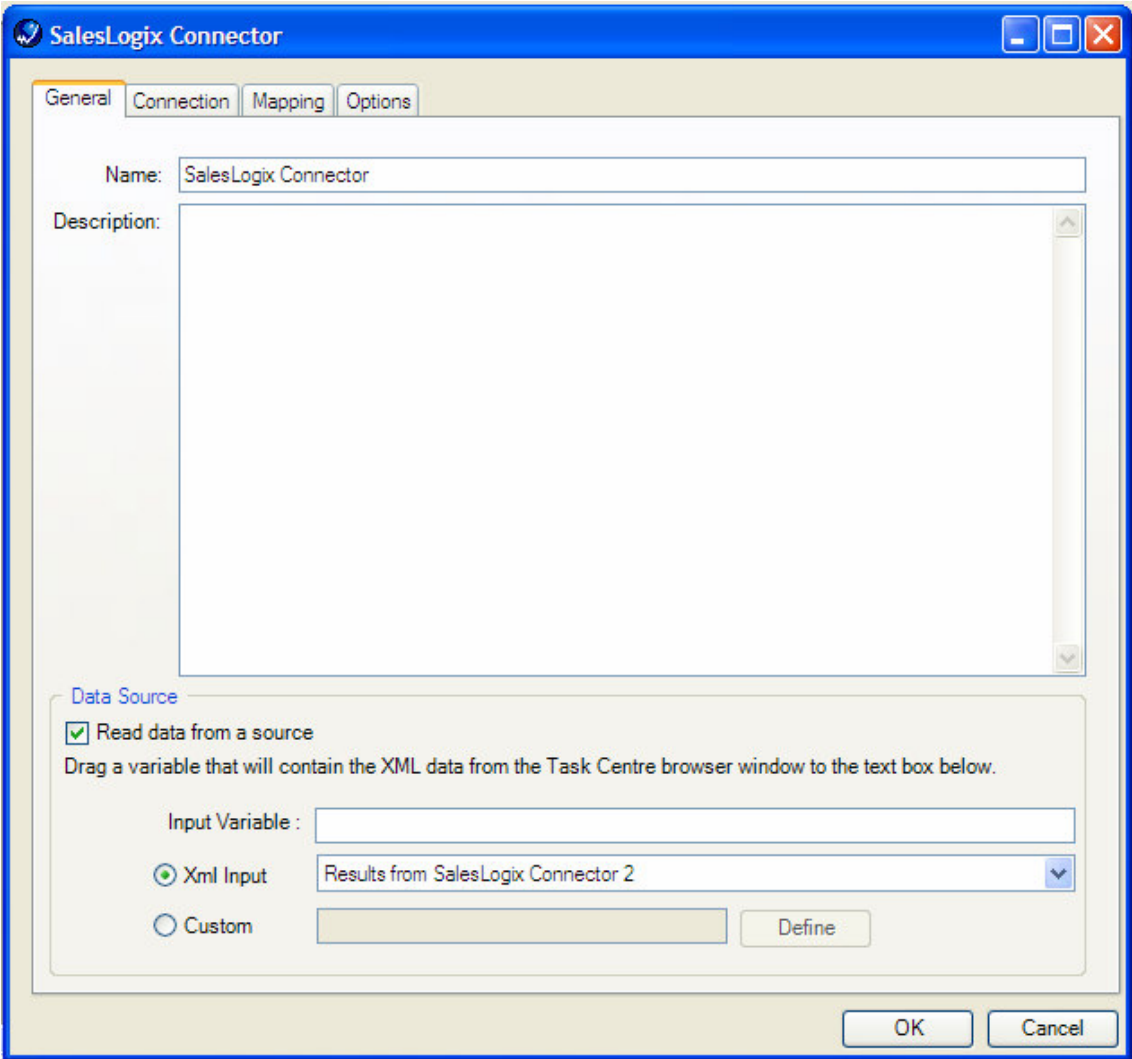


Figure 3 – SalesLogix Connector dialog – General tab

Through the Connection tab of the SalesLogix Connector dialog, see Figure 4 below, a connection is also required for the Step. This is selected from the list of connections previously created and set up in the Global Configuration for the Tool through the Connection Configuration dialog shown in Figure 2.

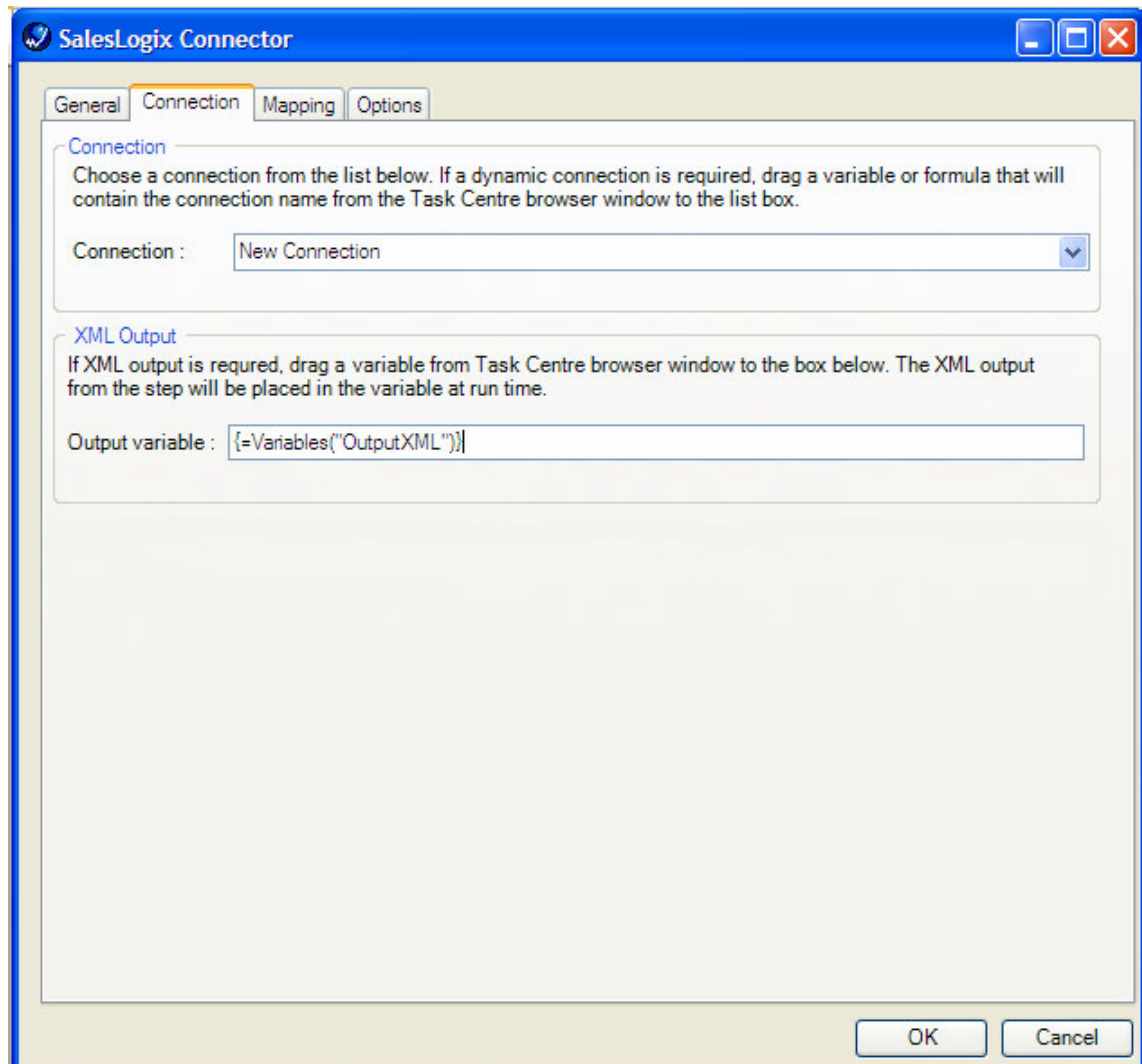


Figure 4 – SalesLogix Connector dialog - Connection tab

As an alternative to selecting a connection that has been previously created, a TaskCentre variable may be utilised by dragging and dropping it from the Task Browser dialog into the Connection field to create a dynamic connection. If the dynamic option is used, then at runtime the text in the variable should match the name of one of the connections set up in the Global Configuration and this connection is then used.

The TaskCentre variable where the XML output is to be placed at runtime is also setup.

Once the input data and the SalesLogix connection have been defined, the mapping of data from the source to a SalesLogix object, along with the operation required, is performed through the Mapping tab of the SalesLogix Connector dialog. See Figure 5 below

The aim of this tab is to define the links between the incoming data and the outgoing data that is to be sent to the Connector Web Service and then into SalesLogix via the DI Server. These links are created by dragging and dropping fields from the left hand box into the right hand box to connect the fields on each tab.

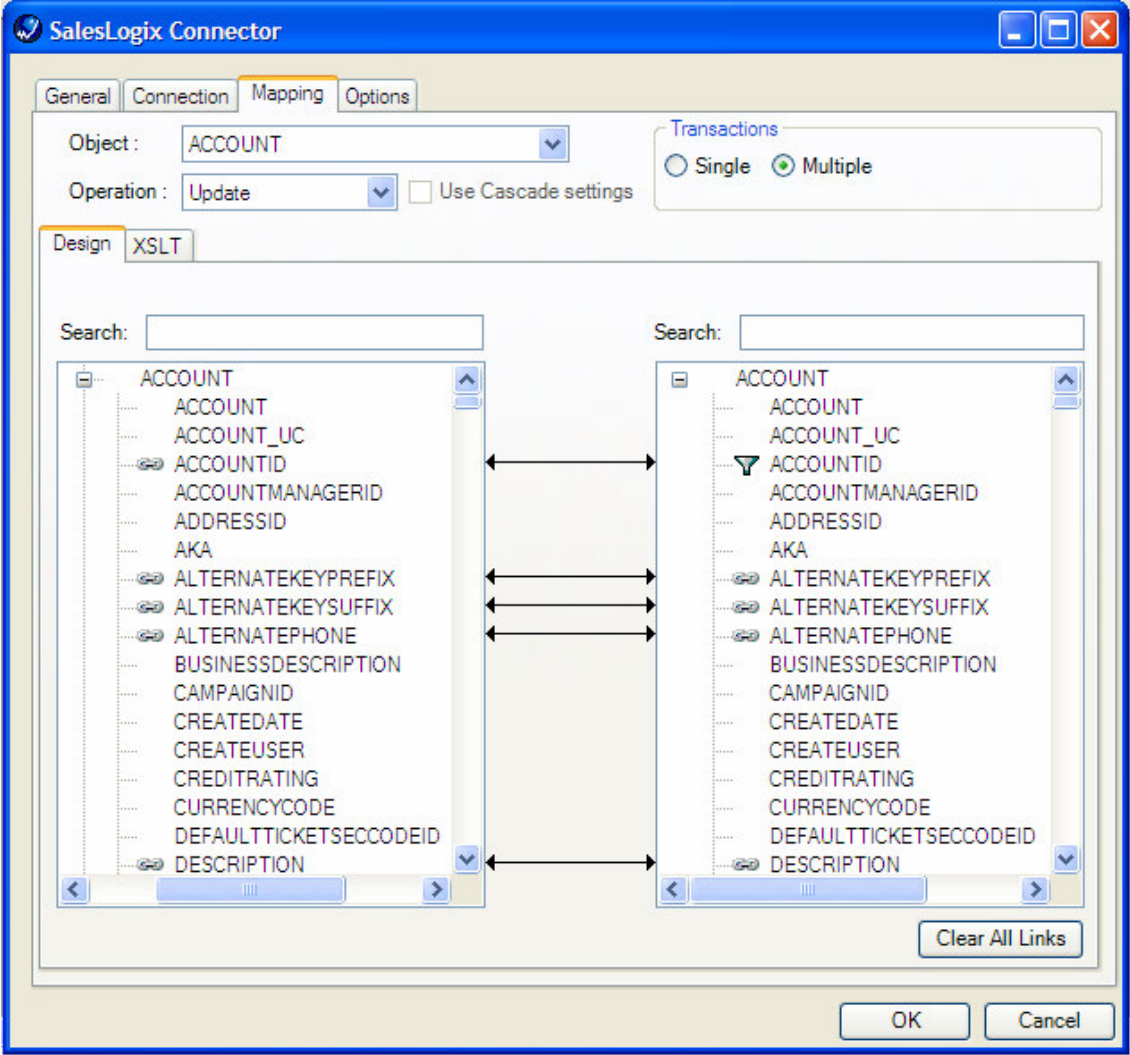


Figure 5 – SalesLogix Connector dialog - Mapping tab

The Object dropdown holds a list of objects available for the selected SalesLogix connection. These are automatically obtained from SalesLogix when the connection is created in the Global Configuration. Selecting an object will change the tree display in the right hand pane to reflect the fields and child objects available for that selected object.

The Operation dropdown displays the operations that the selected object supports. These are Get, Update, Add and Delete.

The aim of this screen is to define the links between the incoming XML data and the outgoing XML data that is to be sent to the Web Service and then into SalesLogix. These links are created

by dragging and dropping fields from the left hand XML tree control to the right hand XML tree control.

TaskCentre variables and formulas can be included in the source data, by simply dragging them from the TaskCentre Browser dialog into a branch node in the XML tree control. Once in place, these can then be linked to the output XML tree control in the same way as normal fields are.

Creating the links in the user interface defines how, at runtime, the incoming XML is to be translated into the XML required for the given object and operation. The method used to translate the XML is XSLT (eXtensible Stylesheet Language Transformations), which is the industry standard mechanism for performing XML translations. There is an XSLT tab on the control, which will show the XSLT currently being generated for the given set of links created. In the case of the user interface not being able to handle a particularly quirky requirement for transforming data, it is possible for the user to go into "Free Type Mode", where XSLT can be input directly to achieve a transformation from the source XML format into the XML format required for the desired object and operation. XSLT knowledge will be needed to do this. The XML schema that the Web Service expects will be published and made available with the Tool

Filter Links

For the update, delete and get operations, the user interface allows the user to designate fields within the destination object to use as the filter when performing the operation. Such fields are linked in the normal way from the source data, but are then marked as a "Filter Link". This means that at runtime, the Tool will compare the value in the source data against the SalesLogix database (the condition for the comparison can be set in the UI, and can be "=", "<", ">" or "!=") when performing the operation. The presence of multiple filter links implies that all conditions must be met on all comparisons.

Examples

Get Operation

On a get operation for an Account, the source dataset definition contains an Account object, with a type of "Prospect" in one of the fields. This field is linked to the Type field in the destination Account object. The link is automatically created as a filter field link as for a Get operation, only filter links are valid – ordinary links which indicate sending of data to SalesLogix are not relevant. The filter condition is set to "=". At runtime, all information for those Accounts is returned from SalesLogix that have a Type set to "Prospect".

Add Operation

For an Add operation, filter links are not relevant. The UI prevents their creation as only data transfer links are relevant.

Update Operation

On an update operation for an account, the source dataset definition contains information to be updated for a specific address within a specific account. Data transfer links are created from fields in the source dataset – they are linked to fields in the Address child object in the destination dataset. A Filter Link is created from the AddressID field in the source to the AddressID field in the destination with the condition set to "=", and another similar Filter Link is created from the AccountID field in the source to the AccountID in the destination Account object. At runtime, for each account object in the dataset, the address that matches the AddressID for the containing AccountId will have its information updated with the data from the source dataset for those fields where a data transfer link was created.

Delete Operation

On a delete operation for an Account, the source dataset definition contains an Account object, with an AccountID field. This is linked to the AccountID field in the destination Account object. The link is automatically created as a filter field link as for a Delete operation, only filter links are valid – ordinary links which indicate sending of data to SalesLogix are not relevant. The filter condition is set to "=". At runtime, all Accounts are deleted from SalesLogix that have an AccountId field set to the same value as the AccountId field for each Account object in the source dataset.

Important Note

Please note that for all operations, SalesLogix security and business logic embedded in the SalesLogix OLE DB provider may affect the intended outcome; for instance selection of the records intended may not be possible due to security restrictions. However, these potential restrictions are unknown until runtime, as the data to be used (as well as, possibly, the connection details to be used) is not known at configuration time

Transaction Control

It is possible to set the transaction configuration for the Step to be "Single" or "Multiple". This controls how the data sent to SalesLogix by the Step is processed. For "Single", all data is processed in a single transaction, so that any error or failure encountered when performing the operation through the SalesLogix OLE DB Driver for any individual record results in the entire operation being failed, so that none of the records are processed. For "Multiple", each record in the dataset is processed individually, so that any error or failure for one of them does not affect the others.

Tool Output

The Tool always exposes a recordset. This has a static format, and has one record per top level object instance that is in the XML sent to SalesLogix. The record contains information on the key field value for the SalesLogix object, along with error messages and codes from the SalesLogix OLE DB provider if the operation for that object instance failed.

The Tool can also output XML and places this XML at runtime into a TaskCentre variable. This XML could contain a dataset returned from SalesLogix (in the case of a Get operation) or the

same information presented in the output recordset, exposed as XML (i.e. a dataset containing the key fields for the object instances affected by the operation. For example, in the case of an update, the key fields for the top level object instances updated would be returned and any error messages if applicable.

The XML can then be used within other TaskCentre Tools, either as is or through conversion to a TaskCentre recordset using the forthcoming XML to Recordset Tool (to be released at the same time as the SalesLogix Connector Tool).

Runtime

The incoming XML dataset is translated into the XML format for the object and operation selected in the configuration. The data for the linked fields is brought across into the output XML – only those fields that were linked are brought across. This XML is passed to the web service which then processes the data, performs the operation requested, and sends back an XML dataset containing the response. This is then assigned to the output variable defined in the step configuration if one was defined. The output Recordset is also created from the XML.

Errors and failures

There are two types of errors that can happen at runtime.

Errors that happen outside of the SalesLogix OLE DB Provider

These types of errors could include:

- XML transformation errors
- Web Service connection errors
- Login to SalesLogix (at initialisation of the OLE DB provider) errors

These types of errors result in the step failing. Whether or not the Task fails depends on the standard Step error configuration options set. The error is shown in the TaskCentre log for the Task and is reported in the message shown in the TaskCentre client.

Errors/warnings/messages that are returned from the SalesLogix OLE DB Provider

These types of errors could include:

- Access permission failures for the given data and login credentials
- Invalid data (according to SalesLogix) passed into the tool for the given fields
- SalesLogix business logic errors

These types of errors will not result in the Step failing. The message from the OLE DB provider is returned in the result XML, which can be written to the TaskCentre event log for the Task. The message shown in the TaskCentre client will reflect that some of the records failed; the result Recordset will give the detail of this.

SalesLogix Connector Web Service

Introduction

The SalesLogix Connector Web Service is the communication layer which enables the SalesLogix Connector Tool to perform operations on data held in SalesLogix databases. It is the only component that directly connects to a SalesLogix database and always connects using the SalesLogix OLE DB driver, never by any other method. This ensures that SalesLogix business rules are adhered to.

Connection information (e.g. server name, database, login name/password) is passed to the web service from the Connector Tool. For each Step and Task instance, the Connector Tool is able to dynamically set the connection used by the SalesLogix OLE DB Driver.

The Web Service performs two vital roles. It is used by the Connector Tool at Step configuration time to create the SalesLogix object schemas, and is also used by the Connector Tool at Task run time to pass data to and retrieve data from a SalesLogix database.

Schema Creation

The Connector Tool works both at configuration and runtime by processing object schemas. These object schemas are obtained from the Web Service. When the Connector Tool, as part of the Global Configuration, requests from the Web Service the available objects in SalesLogix, all the object schemas for the given SalesLogix database are created. For SalesLogix, the word "object" can be interchanged with "table" when referring to entities within the SalesLogix database.

The schemas are created automatically, one per SalesLogix object, by looking at the three tables in the SalesLogix database that store the information for all the available objects, fields and relationships (including extension tables and fields) – RESYNCTABLEDEFS, SECTABLEDEFS and JOINDATA. Each record in RESYNCTABLEDEFS represents one SalesLogix object, whose fields are represented in SECTABLEDEFS and whose relationships to other objects are represented in JOINDATA. From this information, a detailed schema can be created for each object, which includes all fields for that object. Links to other objects are represented by creating a link to that object's schema, where the link information in SalesLogix indicates that it's a child object. The Connector Tool is then able to "walk" the schema, and all the linked schemas, at configuration time in order to create its user interface.

Because all user defined fields and tables created through SalesLogix are stored in these three tables, it means that these are automatically picked up by the schema creation system. The extension tables and fields are created as any other SalesLogix object schema is. The SalesLogix display names for objects and fields, which can be setup in SalesLogix to be whatever is required, are also brought into the schemas, so that the Connector Tool can use these in the user interface.

The schemas also contain information about what operations are supported for the SalesLogix objects. All objects will support Get, Add, Update and Delete and so these operation types will be set in the schemas, meaning that the Connector Tool will pick these up for display in the

configuration user interface. Note, however, that some operations for some fields within an object will be restricted by security settings set for the SalesLogix user selected for the connection details. This is determined by the SalesLogix OLE DB driver at runtime.

All schemas are in industry standard XSD format.

Using SQL views through SalesLogix

In recent versions of SalesLogix, it is possible to create a view within the underlying RDBMS and expose that view in SalesLogix. This view will be brought into the Connector Tool and therefore all operations in theory can be performed upon it. However, caution should be used when attempting anything other than a Get operation, as the exact behaviour that will happen when adding/updating/deleting through a view is dependent on how the view has been set up in the underlying RDBMS.

Data Processing

When a Task containing a SalesLogix Connector Step runs, the Connector Step itself performs all of the processing of data from other Steps in the Task and assembles an XML document containing all of the information that the Web Service needs to perform the operation on the object that the Step is configured for. This information includes the data itself to be passed from other Steps, connection details for the SalesLogix OLE DB provider, details of the database table and fields names that are to be used (available from the schema information) and details of any filter links (with their data and condition values). The XML is then transmitted to the Web Service via an HTTP or HTTPS connection. The web service then parses the XML and performs the required operation by translating it into SQL statements which are then passed to the SalesLogix OLE DB provider. An XML document is then created which contains the response from the SalesLogix OLE DB Provider. This may be a dataset of object data (if the operation was a Get operation, for instance) or could be an error (if, for instance, a Delete operation was attempted on an object that could not be deleted). This XML is then returned to the Connector Tool.

How data is processed for each operation

There are four operations possible when processing data from the Connector Tool.

Add

When adding data to SalesLogix, we are able to take advantage of the system within SalesLogix to automatically create the new key field values for objects. The SalesLogix configuration tables tell us what the key field is for the object being added and also indicates whether or not this key field's value will auto-increment to the next value when an object is added. If this is set to auto-increment we do not take any action as the OLE DB Provider itself will take care of automatically filling this value in when we insert the record.

If not, we call the SalesLogix OLE DB Provider function SLX_DBIDS which gives us the next key field value, which we then use when inserting the record.

When adding child object records (e.g. Address information for an account) we are able to use the new object key field value we obtained when inserting the parent object and can automatically use this as the value for the foreign key field when inserting the child object.

For each data object contained within the request XML, the Data Transfer Link information along with the key field information as outlined above is used to generate an SQL INSERT Statement.

Update

For each data object contained within the request XML, the Filter Link information is used to generate an SQL SELECT Statement. The recordset returned from this is used to power the update operation. SQL UPDATE statements are generated using the key field information in the recordset returned by the SELECT statement, along with the data transfer link values that have been set in that object within the request XML.

Delete

For each data object contained within the request XML, the Filter Link information is used to generate an SQL SELECT Statement. The recordset returned from this is used to power the delete operation. SQL DELETE statements are generated using the key field information in the recordset returned by the SELECT statement. We are intended to build in functionality to emulate the Cascade settings on the join information within SalesLogix – see “Cascading Deletes” for more details on this.

Get

For each data object contained within the request XML, the Filter Link information is used to generate an SQL SELECT Statement. The data returned is transformed into XML before being returned from the Web Service.

Transactions

The SalesLogix OLE DB provider supports transactions. This means that we are able, within the Web Service, to ensure consistency when processing data. For example, an update is required to fields in an object, and fields also within a child object - e.g. a Ticket and a Ticket Solution for that ticket. There will be two SQL UPDATE statements generated, but these can all be performed within the context of a transaction, meaning that if the update to the Ticket Solution table fails for whatever reason, the whole transaction is rolled back including the update to the main Ticket table.

On top of this, the transaction setting from the step configuration is used to wrap either each individual top level record operation in its own transaction, or to wrap all top level record operations up within one transaction.

Data Security

All data passed from the TaskCentre server to SalesLogix is sent via our Web Service over HTTP, as XML data. This includes login details for SalesLogix. To secure this, we will encrypt connection details when they are transmitted from TaskCentre to our web service, then un-encrypt those details within the Web Service before passing them to the SalesLogix OLE DB Provider which expects them un-encrypted. The encryption standard used will be AES (Rijndael), using an asymmetric encryption algorithm. For customers who wish to encrypt all traffic going through the Web Service, HTTPS can be enabled for the Web Service. This will mean the customer having to have a SSL certificate installed on their IIS Server and having the Web Server configuration in IIS setup to use it. All data will then be encrypted.

All SalesLogix OLE DB connection details entered as part of the Connector Tool configuration will be stored encrypted in the TaskCentre store, using TaskCentre's built in encryption.

SalesLogix Security

The SalesLogix OLE DB provider is the only method used to connect to a SalesLogix database and read/write data. This means that all security present in the OLE DB provider is used and also means that any improvements made by Sage to the OLE DB provider in future should be automatically used.

Currently (as of SalesLogix 7.2), the security implementation in the OLE DB provider is NOT the same as the security in the SalesLogix client itself. It is important to understand the differences in the security mechanism available.

When the Global Configuration for the Connector Tool is set, part of the information entered for each connection is the SalesLogix user and password to be used when connecting through the OLE DB provider.

The OLE DB provider will then take the security settings for this user into account when performing SELECT statements (analogous to performing a Get operation in a step). This means that, for instance, if a SalesLogix Connector Step is configured to do a Get operation on an Account object, using a filter field that matches all Accounts in the database, only those accounts that the SalesLogix user defined for the connection used in the Step has permission to see will be returned. Furthermore, if field level security is set on the Account table, any fields that the user does not have permission to view will be returned as NULL. This means that the Connector Tool is able to take into account security set up in SalesLogix for the user configured for the connection when performing a Get operation.

The situation is very different for the remaining three operation types, however. When data is written to SalesLogix (Add, Update or Delete operations), the OLE DB provider does not take into account any security restrictions set up for the SalesLogix user configured for the connection used in the SalesLogix Connector Step. This means that even if the user has got the options to add or delete records set as denied in SalesLogix, through the SalesLogix Connector Step these options are ignored as the OLE DB provider does not take them into account. Similarly, field level security is also ignored by the OLE DB provider.

A SalesLogix Connector Step is therefore able to add, delete or update any object in the database, regardless of the SalesLogix user used in the connection configuration for the Step. This is important to understand.

It is possible to create a SalesLogix custom security provider that is used to augment the security provided by the OLE DB Provider. It is believed that the Connector Tool's use of the OLE DB provider should take any custom security provider into account when performing an operation. This is to be tested and determined.

It is theoretically possible for the SalesLogix Connector Tool to have its own functionality developed which duplicates the security provided by the SalesLogix client. This functionality would exist outside of the OLE DB provider. This is not desirable for a number of reasons, most importantly being that it would introduce problems if Sage change the way, even slightly, that security is implemented in either the SalesLogix client or the OLE DB provider. It has been decided that the SalesLogix Connector Tool will not try and add to the security mechanisms in place in the OLE DB provider. Any future improvements to the security mechanism in the OLE DB provider should be picked up automatically by the Tool.

We will support the use of the SalesLogix connection Read/Write password. If a Read/Write password has been set on the connection from within the SalesLogix Connection Manager, then this can be used by the Tool by setting it as an option in the Global Configuration for a connection.

SECCODEID

When adding objects through the SalesLogix Connector Tool, the SECCODEID field has to be set for those objects that have this field. This would mean mapping a value to this field on every Step that adds an object. We will provide an option on the Global Configuration dialog for a connection to set a default SECCODEID value that will be used wherever a SECCODEID field is present on an object that is to be added, and where the Step configuration itself has not defined that value through the use of a mapping from a data source. This is simply a mechanism to make adding objects a little easier for situations where the SECCODEID value will always be the same for every addition.

Cascading Deletes

When defining a table join in SalesLogix, the user sets up the cascade deletion rule for that join. This information is stored in the JOINDATA table and so can be picked up when the object schemas are created. This means that if a delete operation is performed on an object, the cascade setting can be taken into account when performing the delete operation; the related child objects can be processed according to the cascade setting set for the join in SalesLogix. However, the Replace option won't be supported. A join with this option set will simply act as if it is set to the Clear option and the foreign key values will be set to NULL.

The OLE DB Provider itself does not offer this functionality, but the SalesLogix Connector Tool can implement this itself. It is intended to make the use of the cascade settings optional on a per Step basis. If the cascade settings are not used, then any child records will simply be left alone (much like the Ignore option for a join in SalesLogix).

Synchronisation

The SalesLogix OLE DB provider implements synchronisation support. This means that add/update/delete operations performed through the SalesLogix Connector Tool will be automatically logged for synchronisation.

Attachments

Attachments in SalesLogix are represented in the Attachments table within the database. There is no SalesLogix mechanism available for adding attachment files to SalesLogix from outside of the SalesLogix client; thus ensuring that the attachments themselves are available for synchronisation. If it is necessary to add an attachment, the file will have to be placed in the correct directory, and given the correct name, by other methods. The filename used can then be mapped in the SalesLogix Connector Tool through to a new Attachment record if necessary.