orbis software
THE POWER TO MAKE IT HAPPEN

Sage CRM Connector Tool
White Paper

About the Sage CRM Connector Tool

For further information, please contact your TaskCentre distributor.

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

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.

Introduction

The Sage CRM Connector Tool gives a simple, straightforward way of providing connectivity between TaskCentre and Sage CRM. Without this Tool, it is possible for TaskCentre to read data from a Sage CRM system via an OLE DB or ODBC Database Query Tool. However, it is not straight forward to write data to Sage CRM via TaskCentre and large amounts of script are needed to achieve this. The Sage CRM Connector Tool takes away the need for script and provides an easy to use user interface for writing data to Sage CRM. Another advantage of the Sage CRM Connector Tool is that it writes data through the Sage API. This means that the data is validated using the business rules defined by Sage so the data cannot be accidentally corrupted through lack of knowledge of these rules.

The Sage CRM Connector Tool uses XML as its input and output data format. Interoperation with standard TaskCentre Tools is achieved by the use of the TaskCentre XML Tools which allow conversion of standard TaskCentre recordsets to/from XML.

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

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

System Requirements

Hardware

There are no special requirements to use this Tool. Please refer to the TaskCentre system requirements on the Orbis Software website.

Software

The software requirements to use the Sage CRM Connector Tool are:-

- Sage CRM Version 6.0 and above.
- TaskCentre 4.5 Service Release 3 and above.
- Microsoft .Net Framework Version 3.5 or higher

The software requirements to use the Sage CRM Web Service are:-

- Microsoft IIS Version 6 or higher.
- Microsoft ASP.Net
- Microsoft Windows 2003 or higher.
- Microsoft .Net Framework Version 3.5 or higher.

.

Architecture

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

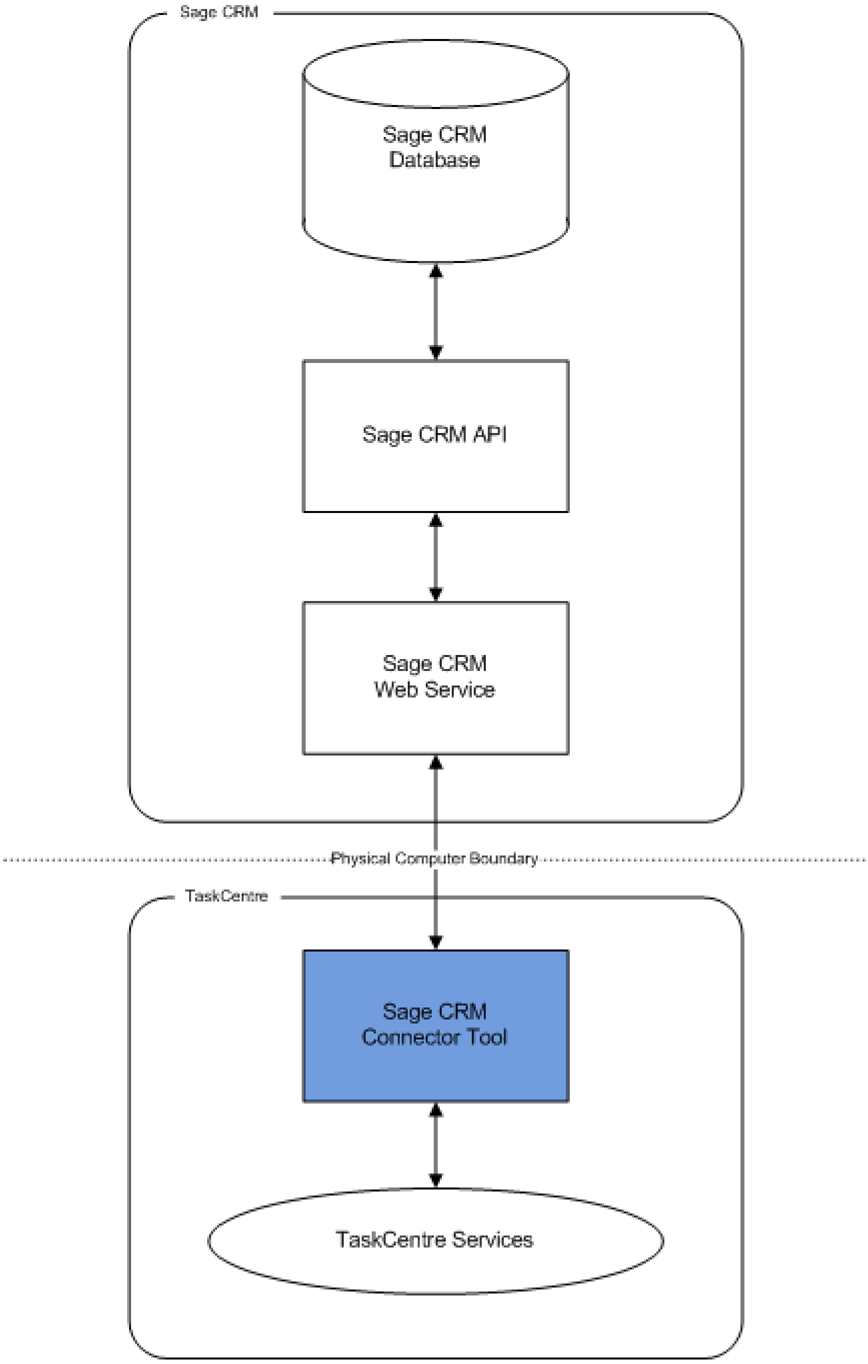


Figure 1 – System Architecture Diagram

Sage CRM Connector Tool

Introduction

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

The Tool transforms incoming XML data into an XML format for the specific Sage CRM object and operation selected. This XML is then passed to the Connector Web Service which does the actual communication with Sage CRM via the API. The response is exposed as XML from the Tool, which could include details of the object requested, errors from the provider, or the values of any automatically generated ID fields.

The Tool does not accept a TaskCentre recordset directly as input but TaskCentre variables containing XML data from any source at Task runtime may be utilised. In addition, a TaskCentre recordset can be processed by the Recordset to XML Tool first to transform it into XML if you have this Tool available.

Technical Summary

Working with other Steps

The Sage CRM Connector Tool is able to expose information to any other TaskCentre Tool that accepts an XML input. This could be another Sage CRM Connector Tool or the XML to Recordset Tool, which converts the information into a TaskCentre recordset for use with other TaskCentre Tools. It consumes information in the form of an XML input, which can come from any Tool that is capable of outputting XML. It can also consume information in the form of XML data contained within a Task variable.

Objects Consumed

Objects that the Sage CRM Connector Tool consumes that are exposed by other Steps are listed below.

- **Task Variable** - XML data is extracted from the designated Variable. This XML data can come from any source, as long as it has been placed into a nominated Task variable at runtime. If choosing this option, at configure time the schema of the XML to be used at runtime must be set.
- **XML** - XML data from any TaskCentre Tool capable of exposing XML. This is exposed through the Task Browser dialog in the 'InputData' object.

Objects Exposed

Objects that the Sage CRM Connector Tool exposes that can be utilised by other Steps are listed below.

- **InputData** - The Sage CRM Connector Tool places the XML input data used by the Step in a property called 'InputData', which can be used by other Tools.
- **OutputData** - The Sage CRM Connector Tool also places the 'success' XML output from the Step in a property called 'OutputData', which can be used by other Tools.

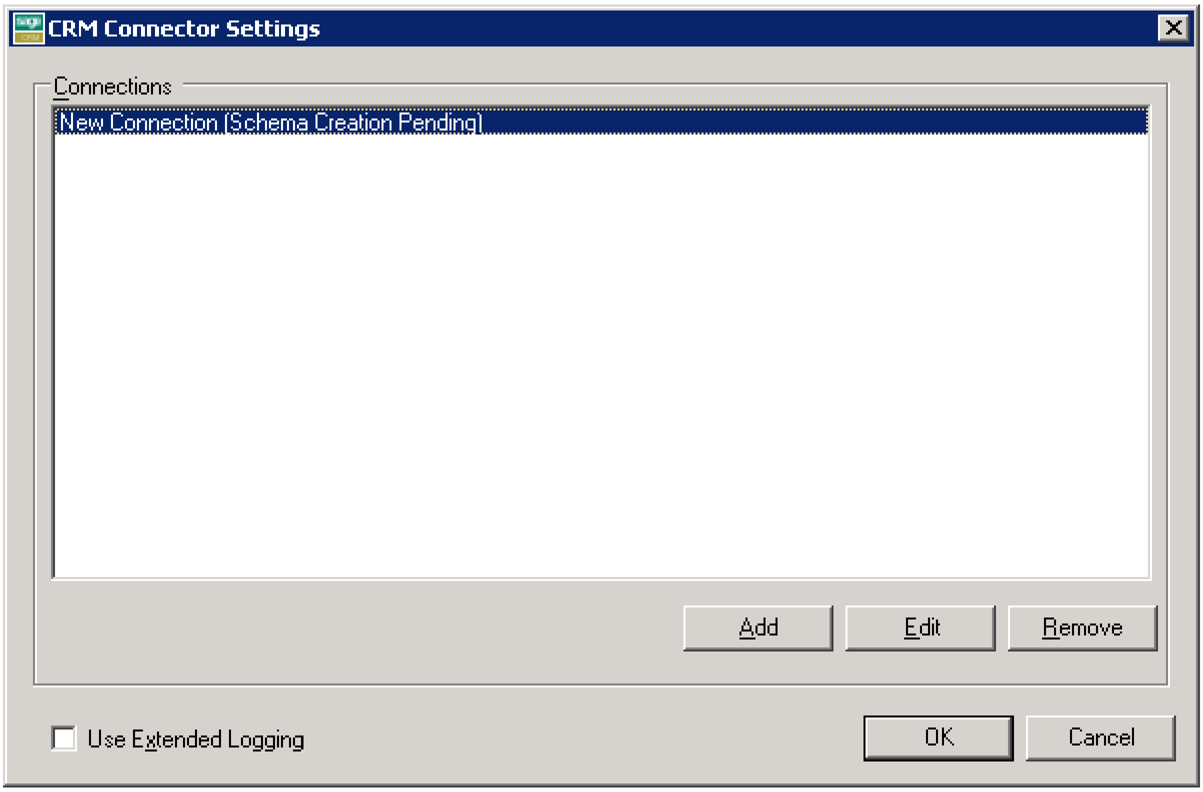
- **ErrorData** - The Sage CRM Connector Tool also places the 'error' XML output from the Step in a property called 'ErrorData', which can be used by other Tools.

Properties Exposed

- **RunCount** - At runtime, this contains the total number of top level objects in the input XML that were processed by the Tool.
- **FailCount** - At runtime, this contains the number of error objects returned in the output XML.
- **SuccessCount** - At runtime, this contains the number of non-error objects returned in the output XML.
- **OutputXMLString** - The 'OutputData' from the Step is exposed in a string, which can be used by other Tools.
- **ErrorXMLString** - The 'ErrorData' from the Step is exposed in a string, which can be used by other Tools.
- **ResponseXMLString** - The response received from the Sage CRM API is exposed in a string, which can be used by other Tools.

Global Configuration

The global configuration dialogs for the Tool, shown in Figures 2 and 3 below, are where connections to the Sage CRM Web Service, which communicates with Sage CRM, are defined.



CRM Connector Settings

Connections

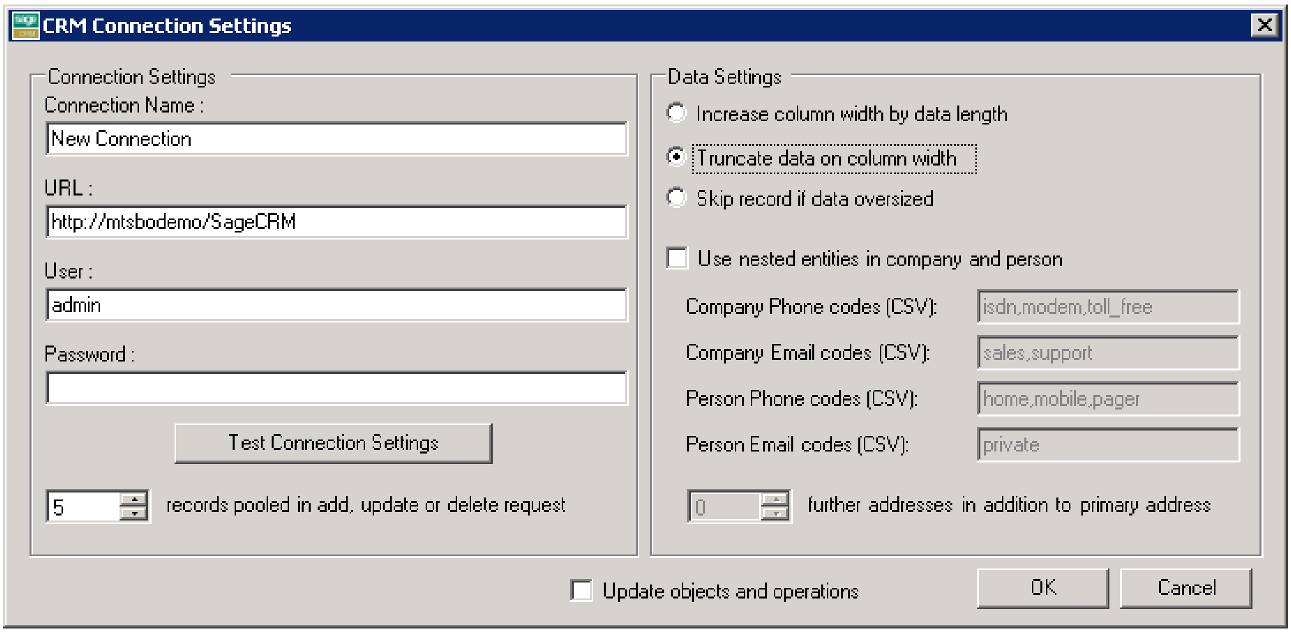
New Connection (Schema Creation Pending)

☐ Use Extended Logging

Figure 2 – Sage CRM Connector Tool – Global Settings dialog.

The information for each connection includes:

- **Connection Name** - Name for the connection.
- **Web Service URL** – The Web Service URL to be used for the connection.
- **User** - The Windows user to be used for the connection. This Windows user must have been mapped onto a Sage CRM user through the Sage CRM API configuration tool. Note also that the Sage CRM user must be enabled to use the UIG formset.
- **Password** - The password for the Sage CRM user to be used for the connection.
- **Data Settings** – A number of options are available for handling input data.


CRM Connection Settings

Connection Settings

Connection Name : New Connection

URL : http://mtsbodemo/SageCRM

User : admin

Password :

5 records pooled in add, update or delete request

Data Settings

○ Increase column width by data length

● Truncate data on column width

○ Skip record if data oversized

☐ Use nested entities in company and person

Company Phone codes (CSV): isdn,modem,toll_free

Company Email codes (CSV): sales,support

Person Phone codes (CSV): home,mobile,pager

Person Email codes (CSV): private

0 further addresses in addition to primary address

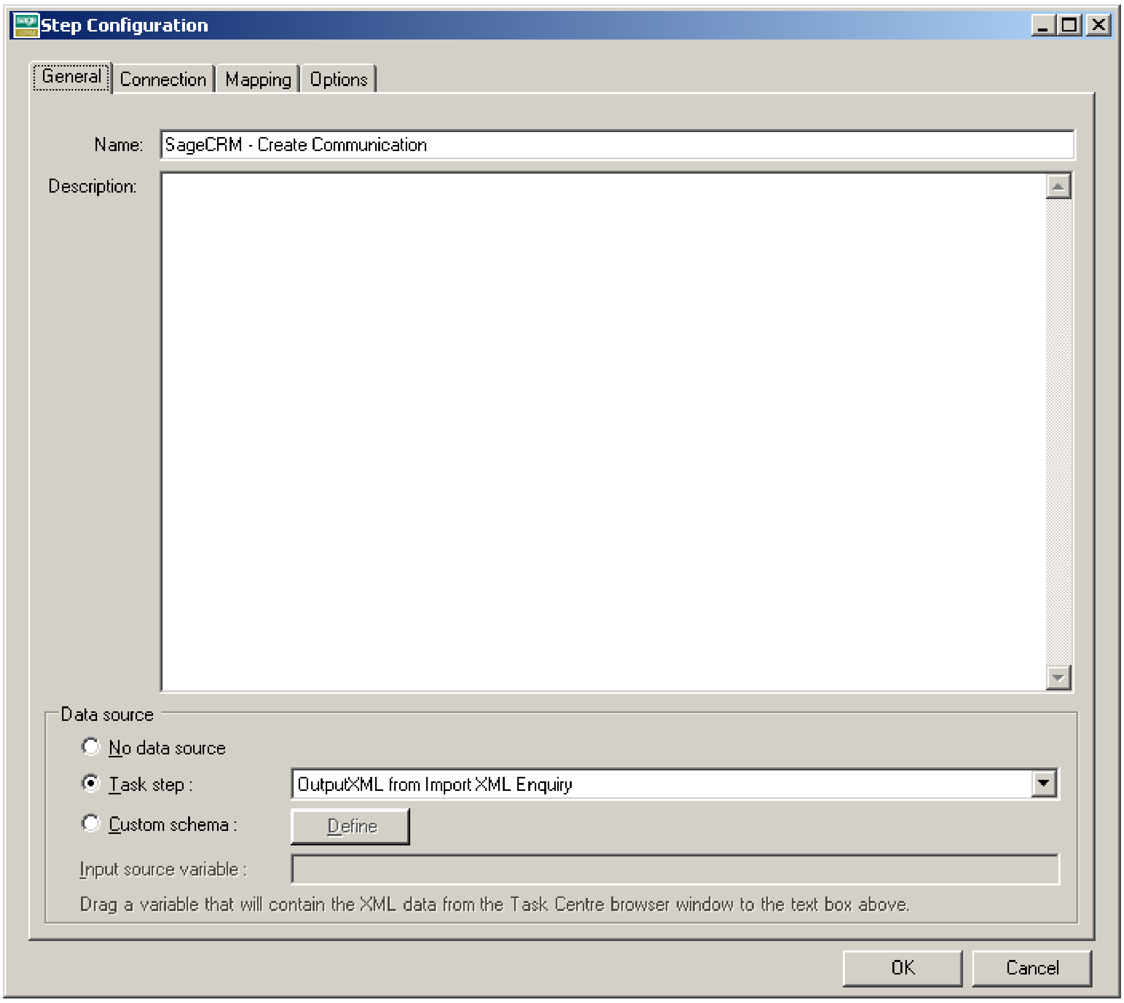
☐ Update objects and operations

Figure 3 – CRM Connection Settings dialog.

Step Configuration

General tab

At configuration time, when the Tool is added to a Task the Step will allow the user to select the data source which it will use at runtime through the General tab of the Sage CRM Connector Tool dialog. See Figure 4 below.



Name: SageCRM - Create Communication

Description:

Data source

○ No data source

● Task step : OutputXML from Import XML Enquiry

○ Custom schema :

Input source variable :

Drag a variable that will contain the XML data from the Task Centre browser window to the text box above.

Figure 4 – Sage CRM Connector dialog – General tab

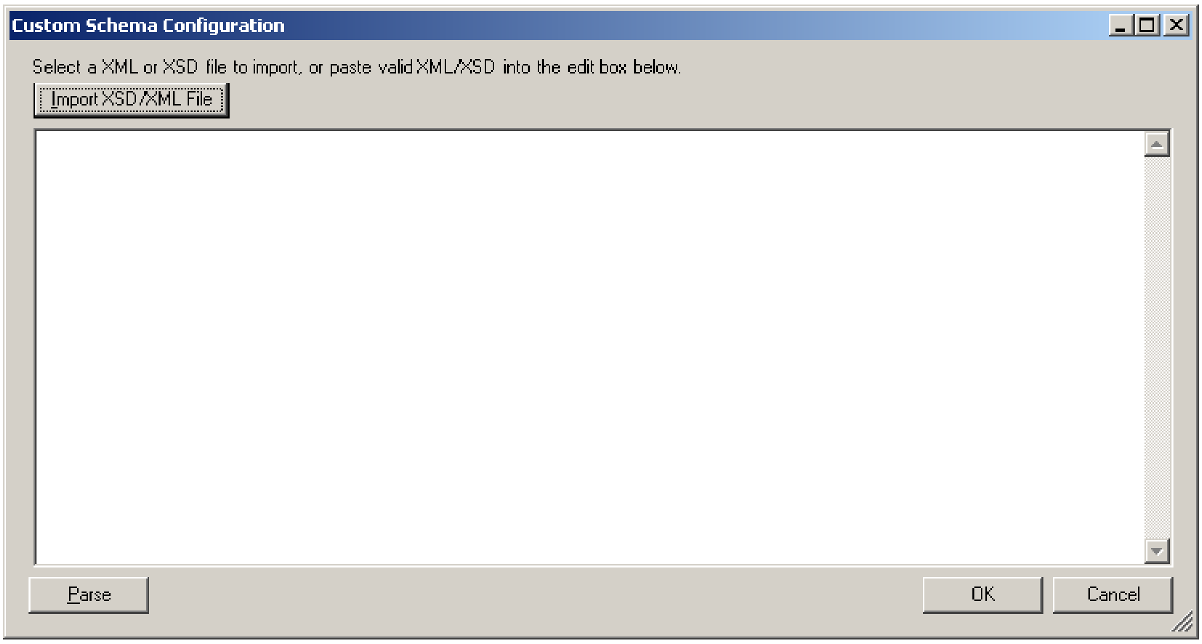
The data source can be either a Task Step, or a custom data source, including the use of TaskCentre variables.

If a Task Step is used, then the data source can be set to an available TaskCentre XML data source. Only those steps that are capable of natively exposing an XML document at runtime are listed (this may be another Sage CRM Connector Step or a Tool such as Recordset to XML).

It is also possible to define a custom data source. This data source has two parts – a definition of the schema used by the incoming data, and the selection of a TaskCentre variable that will contain the XML data at runtime.

Custom Schema Configuration

The Custom Schema Configuration dialog (Figure 5) is used to define the schema of the XML to be used as the input data source for the Sage CRM Connector Step. An XML schema defines the structure of the XML, in terms of what tags are present, and the nesting of the tags. The format used by the Sage CRM Connector Tool is the industry standard XSD format.



Custom Schema Configuration

Select a XML or XSD file to import, or paste valid XML/XSD into the edit box below.

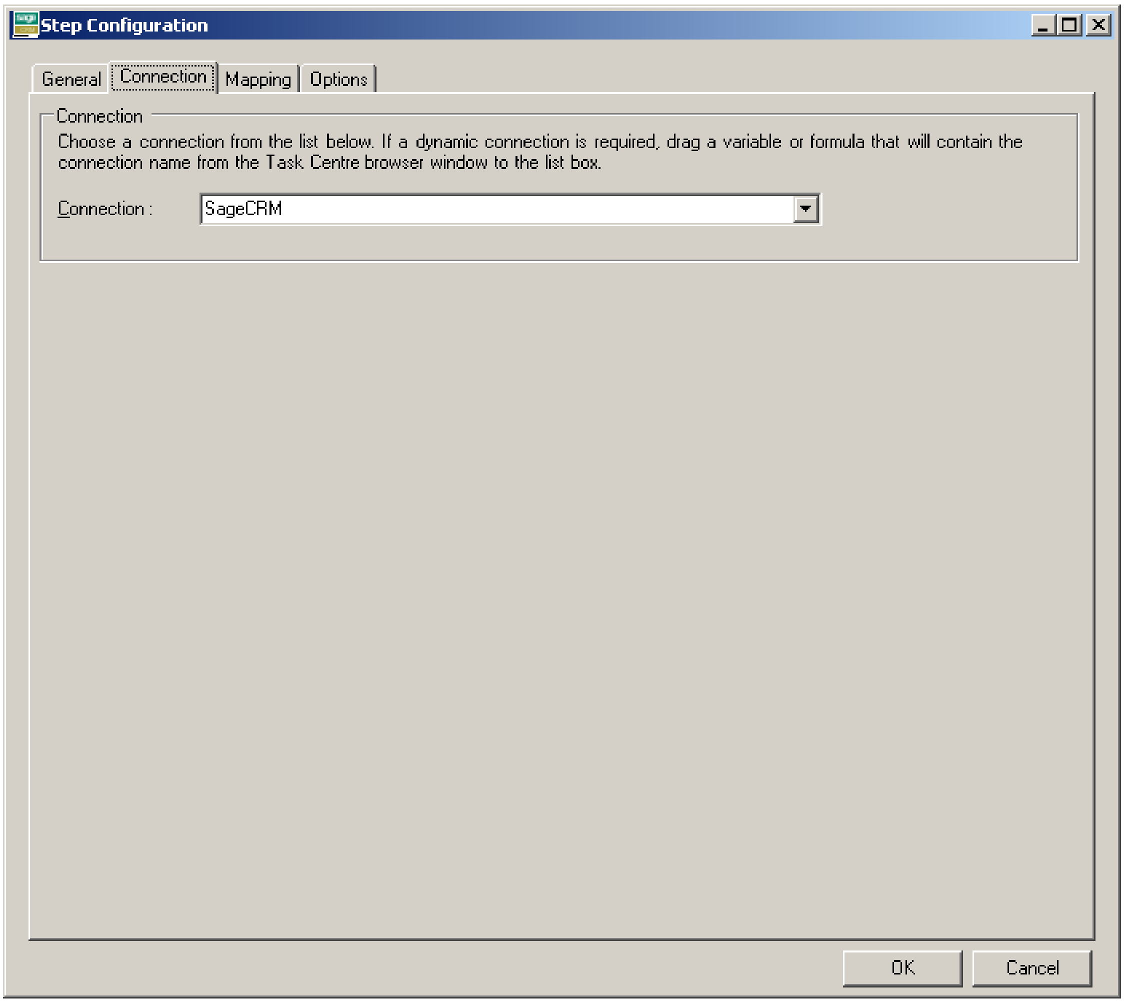
Figure 5 – Sage CRM Connector Custom Schema Configuration Dialog

If an XSD schema file is available for the XML that is to be used as the input data, then this can be loaded in from a file on disk (via the "Import XSD/XML File" button, which displays a file selection dialog) or pasted directly into the text box shown. Any XML passed into the step at runtime must conform to this schema. Otherwise an error will be reported.

Using an XSD schema file is the recommended way of defining the XML schema. However, if an XSD schema is not available, then an example of the XML that will be used at runtime can be loaded in or pasted into the text box.

Connection tab

Through the Connection tab of the Sage CRM Connector dialog, see Figure 6 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 CRM Connection Settings dialog shown in Figure 3.



Connection

Choose a connection from the list below. If a dynamic connection is required, drag a variable or formula that will contain the connection name from the Task Centre browser window to the list box.

Connection : SageCRM

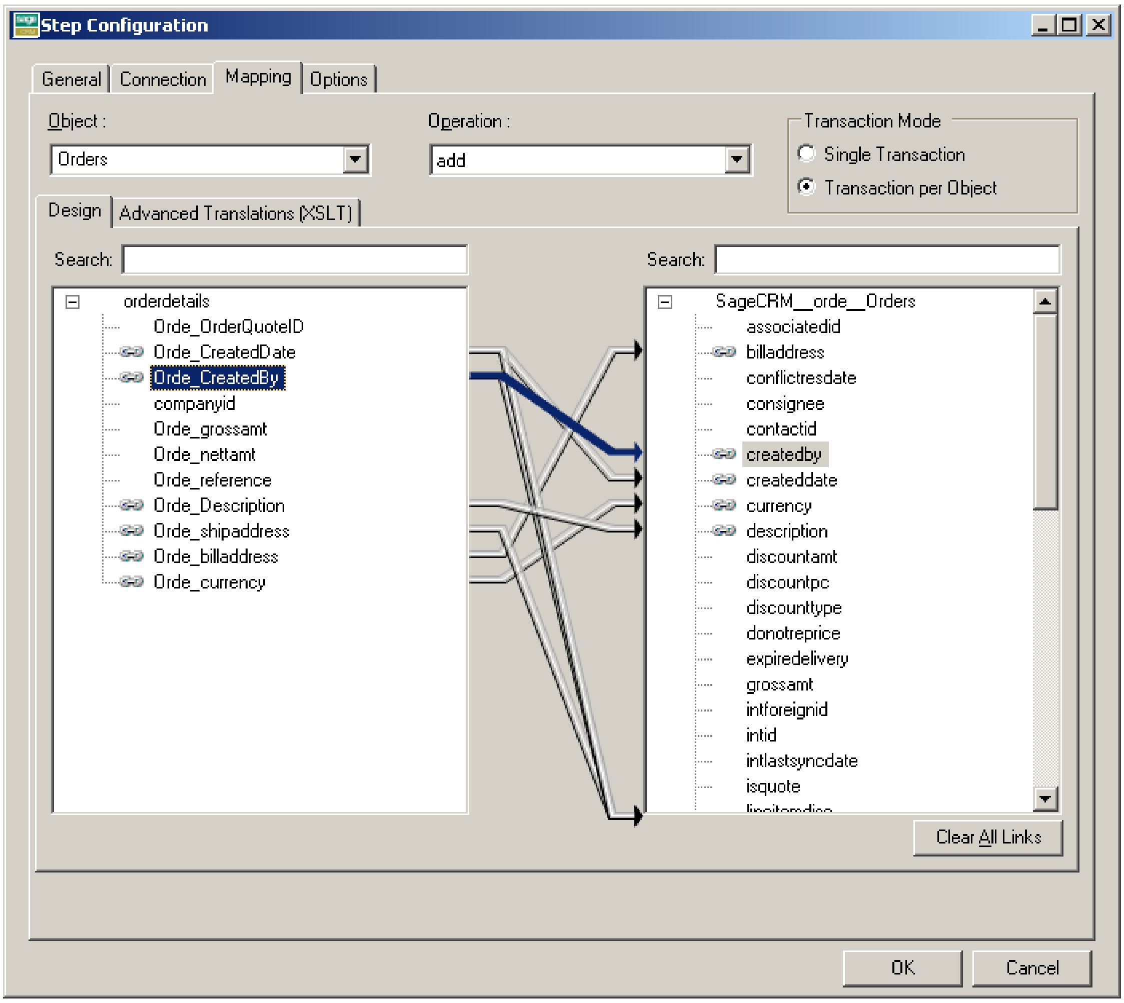
Figure 6 – Sage CRM Connector dialog - Connection tab

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.

Mapping tab

Once the input data and the Sage CRM connection have been defined, the mapping of data from the source to a Sage CRM object, along with the operation required, is performed through the Mapping tab of the Sage CRM Connector dialog. See Figure 7 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 Sage CRM via the Sage CRM API.



Object : Orders

Operation : add

Transaction Mode: ○ Single Transaction ● Transaction per Object

Figure 7 – Sage CRM Connector dialog - Mapping tab

The Object dropdown holds a list of objects available for the selected Sage CRM connection. 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.

All of the objects available for version 6.0 of Sage CRM are included and are shown in the tables below.

Version 6 Objects	**Operation**
Cases	Get Add Update UpdateORAdd Delete
Communication	Get Add Update UpdateORAdd Delete
Company	Get Add Update UpdateORAdd Delete
Lead	Get Add Update UpdateORAdd Delete
Opportunity	Get Add Update UpdateORAdd Delete
Orders	Get Add Update UpdateORAdd Delete
Person	Get Add Update UpdateORAdd Delete
Quotes	Get Add Update UpdateORAdd Delete
Address	Get Add Update UpdateORAdd Delete
Comm-Link	Get Add Update UpdateORAdd Delete
Currency	Get Add Update UpdateORAdd Delete
Email	Get Add Update UpdateORAdd Delete
NewProduct	Get Add Update UpdateORAdd Delete
Notes	Get Add Update UpdateORAdd Delete
OpportunityItem	Get Add Update UpdateORAdd Delete
OrderItems	Get Add Update UpdateORAdd Delete
Phone	Get Add Update UpdateORAdd Delete
Pricing	Get Add Update UpdateORAdd Delete
PricingList	Get Add Update UpdateORAdd Delete
ProductFamily	Get Add Update UpdateORAdd Delete
QuoteItems	Get Add Update UpdateORAdd Delete
Uom	Get Add Update UpdateORAdd Delete
UomFamily	Get Add Update UpdateORAdd Delete
Users	Get Add Update UpdateORAdd Delete

The Operation dropdown displays the operations that the selected object supports. These are Get, Add, Update, UpdateOrAdd and Delete and any special methods available for the selected object.

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 Sage CRM.

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 input 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 Advanced Translation (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 'Enable Free Type Mode' by checking this option, 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.

Filter Links

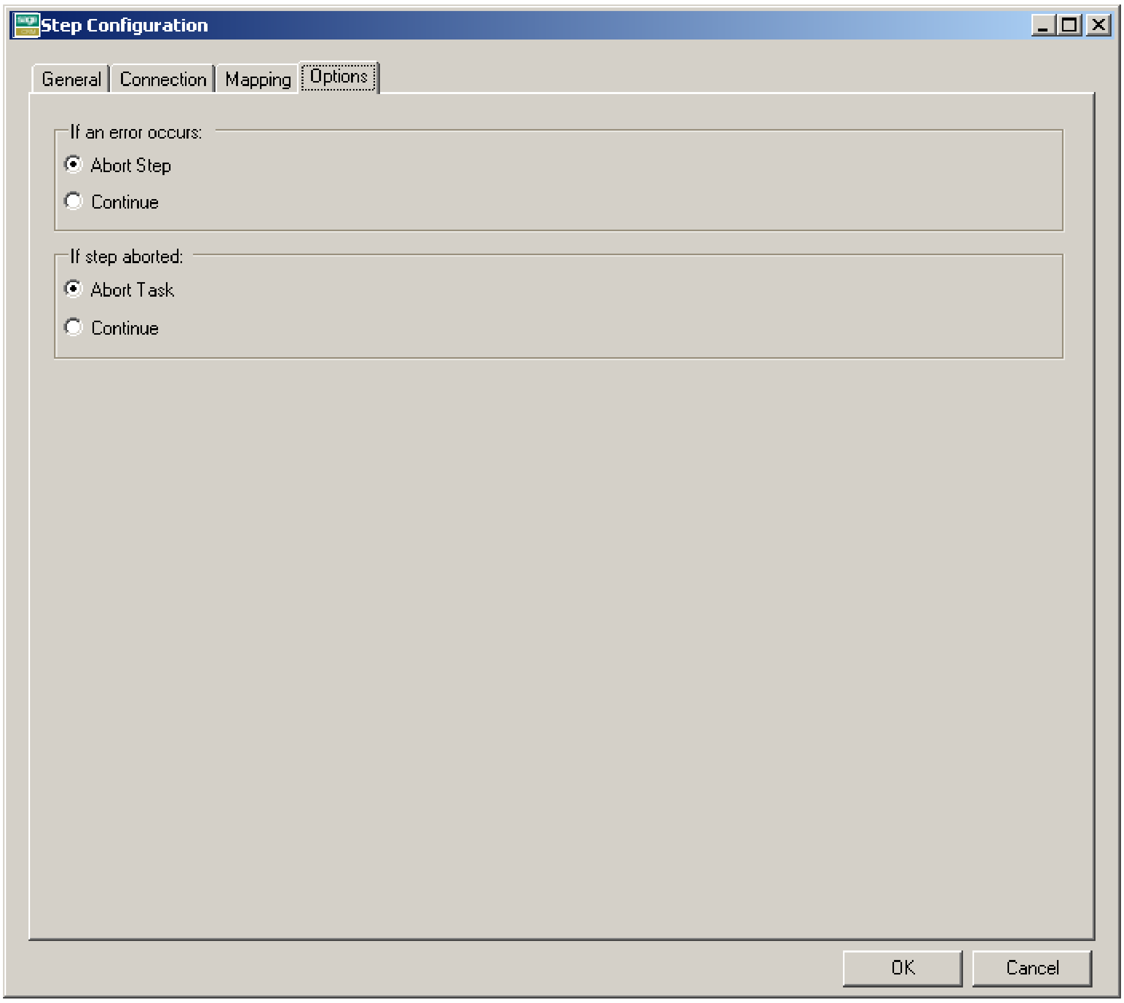
For the Update, Delete and Get operations, the user interface allows the user to designate fields within the destination object to use as the filter when performing the operation. Such fields are linked in the normal way from the source data, but are then marked as a "Filter Link". This means that at runtime, the Tool will compare the value in the source data against the Sage CRM 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.

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 Sage CRM 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 Sage CRM API 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 XML is processed individually, so that any error or failure for one of them does not affect the others.

Options tab

Once the data mapping process has been completed, you can configure the Step behaviour if an error occurs when the Step is being processed through the Options tab. See Figure 8 below.



If an error occurs:

● Abort Step

○ Continue

If step aborted:

● Abort Task

○ Continue

Figure 8 – Sage CRM Connector dialog - Options tab

The following options are provided for selection:

- **'If an error occurs'** (area) – This area contains the following options:
  - **Abort Step** - If there is an error mapping a row from the input data source then the Step will abort. This is the default selection.
  - **Continue** - The Step will continue even if there is an error mapping an individual row from the input data source.
- **'If step aborted'** (area) – This area contains the following options:
  - **Abort Task** - If the Step is aborted then the Task will be aborted. This is the default selection.
  - **Continue** - If the Step is aborted then the Task will continue onto the next Step in the process if one exists.

Output XML

The Sage CRM Connector Tool may output two separate XML documents depending on the Transaction Mode selected in the Mapping tab. The contents for each document type are described below.

If the Transactions per Object mode is selected then potentially both an 'OutputData' and an 'ErrorData' XML document will be produced; each containing the successful or error objects respectively.

If the Single Transaction mode is selected then either an 'OutputData' or 'ErrorData' XML document will be produced. An 'ErrorData' XML document is always produced if an error is returned by the API.

OutputData XML document

This document will contain the following:

- The input data mapped for the object
- Data returned from Sage CRM, in the case of a 'Get' method
- The key fields for the entity instances affected by the method. For example, in the case of an 'Add', the key fields for the top level object instances created would be returned.

ErrorData XML document

This document will contain the following:

- Any error that occurs is created as an <ERROR> node.

  The sub-nodes of the ERROR node are:

  - <Object> - The name of the requested object (i.e. Orderlines.Orlt_quantity)
  - <CODE> - The error code returned by Sage
  - <MESSAGE> - The error message
  - <EXTENDEDINFO> - A string that contains the applicable Sage Column Name, Sage Location Code and Sage Row ID
  - <INPUTDATA> - The input data mapped for the object.

Each XML document output by the Tool can then be used within other TaskCentre Tools, either as is or through conversion to a TaskCentre recordset using the XML to Recordset Tool.

Runtime

The incoming XML 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 document containing the response. The XML can then be used by other TaskCentre Steps.

Error Handling

It is of course possible that errors may happen when the Tool is processing data. These errors could come from a variety of sources and for a variety of reasons, including:

- Web service connection errors
- User privilege errors
- Errors from the Sage CRM API (see Tool Output above)
- Warnings/Messages from the Sage CRM API (see Output XML above)

If a Web Service connection runtime error occurs then the Step will make 3 attempts, at 10 second intervals, to reconnect before writing the error to the TaskCentre event log. It will then follow the rules defined on the Options tab.

All other error types are written directly to the TaskCentre event log.

Sage CRM Web Service

Introduction

The Sage CRM Web Service provided by Sage is the communication layer which enables the Sage CRM Connector Tool to perform operations on data held in Sage CRM databases. It is the only component that directly connects to a Sage CRM database and always connects using the Sage CRM API, never by any other method. This ensures that Sage CRM business rules are adhered to. The Microsoft .NET framework version 2 SP1, IIS and ASP.NET are required to run this.

Connection information (e.g. login name/password) is passed to the web service from the Sage CRM Connector Tool. For each Step and Task instance, the Tool is able to dynamically set the connection used by the Sage CRM API.

The Sage CRM Web Service is used by the Sage CRM Connector Tool at Step configuration time to create the Sage CRM object schemas and is also used by the Sage CRM Connector Tool at Task run time to pass data to and retrieve data from a Sage CRM database.

Schema Creation

The Sage CRM Connector Tool works both at configuration and runtime by processing object schemas. These object schemas are obtained from the Sage CRM Web Service.

When the Sage CRM Connector Tool, as part of the global configuration, requests the available objects in Sage CRM from the Sage CRM Web Service, all the object schemas for the given Sage CRM database are created. For Sage CRM, the word “object” can be interchanged with “entity” when referring to the Sage CRM API documentation. Where an entity has multiple roles in the API, e.g. a purchase order line has the roles comment, service and product these are included as separate objects in the connector.

A detailed schema is created for each object and operation, which includes all fields for that object and operation. Links to other objects are represented by creating a link to that object’s schema, where the link information in Sage CRM indicates that it’s a child object. The Sage CRM Connector Tool is then able to “walk” the schema, and all the linked schemas, at configuration time in order to create its user interface.

All schemas are in industry standard XSD format.

Data Processing

When a Task containing a Sage CRM 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 Sage CRM API, 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 generating a function call to the Sage CRM API. An XML document is then created which contains the response from the Sage CRM API. This may be information about the requested object data if for instance the operation was a Get operation, or it could be an error if a Delete operation was attempted on an object that could not be deleted. This XML is then returned to the Sage CRM Connector Tool.

Transactions

The Sage CRM API 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. an Invoice Header and an Invoice Line for that invoice. There will be two function calls made to the Sage CRM API, but these can all be performed within the context of a transaction, meaning that if the update to the Invoice Line fails for whatever reason, the whole transaction is rolled back, including the update to the Invoice Header.

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 Sage CRM is sent via Sage's own Sage CRM Web Service as XML data, over HTTP. This includes login details for Sage CRM. To secure this, the connection details are encrypted when they are transmitted from TaskCentre to the Web Service and are then un-encrypted within the Web Service before being passed to the Sage CRM API.

The encryption standard used is 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 an 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 Sage CRM API connection details entered as part of the Sage CRM Connector Tool configuration are stored encrypted in the TaskCentre store, using TaskCentre's built in encryption.

Sage CRM Security

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