



Concentrix
a common centre

Microsoft Dynamics CRM v4 Introduction to Custom Automation

Concentrix Limited

20 Granite Way
Mountsorrel
Loughborough
LE12 7TZ

E: info@concentrix.co.uk
T: +44 (0)1509 410500
F: +44 (0)1509 410501

www.concentrix.co.uk

Registered in England No. 3891450
VAT Registration No. 738 0929 09



Copyright © Concentrix Ltd 2009 - 2010

All rights reserved. No part of this document may be reproduced, distributed, stored in a retrieval system or transmitted in any form or by any means electronic, mechanical, photocopying, in writing, recording or otherwise without the prior express permission of Concentrix Ltd.

Concentrix Ltd cannot be held liable for any errors or omissions in this document. However, if you discover any inaccuracies or have suggestions for improving the document, please contact Concentrix directly.



Purpose

The purpose of this document is to provide insight into the options for customising [Microsoft Dynamics CRM](#) beyond adding entities, attributes and relationships.

This document is aimed at Microsoft Dynamics CRM project team members with either a technical or business interest in the solution. For a broader overview of the product, or for any other information about Microsoft Dynamics CRM, please [contact Concentrix](#).



1 Microsoft Dynamics CRM Automation

Microsoft Dynamics CRM supports a number of ways of automating processes and the user experience. Each of these works in a different way and can be used to achieve either different outcomes or the same outcomes in different ways.

It is recommended that each of the methods below is reviewed to ensure that the method chosen is suitable for the requirement:

1.1 Client Side Processing

Client side code occurs actually on the user's machine and often utilises built in functionality provided by Outlook or Internet Explorer. Client side code often occurs in line with the user working with the system and therefore the user is often aware that the processing has occurred.

1.1.1 JavaScript

Client side code is placed within a specific CRM form and is triggered as a result of a manual action taken on a form. (I.e. when a form is loaded, when a user changes a field value, or when a user clicks 'Save').

The code will only fire when changes are made to the form and will not occur as a result of server side changes or updates.

An **example** of JavaScript that may be configured:

When a user selects that a contact's salutation is 'Mr', the gender field is changed to 'Male'.

<Or>

When a start date is populated the end date is automatically populated to be 1 year later.

1.2 Server Side Processing

Server side processing occurs on the server and does not tend to require resources on the client machine. Typically server side code is more powerful than client side code but does have the disadvantage that it can sometimes not be as obvious to the user that the processing has begun.

1.2.1 Workflows

Workflow rules represent server side automated processes that can either be triggered manually or automatically as a result of a programmatic action being taken on a specific record (i.e. when an account is created or updated). When there are available manual workflow rules, there is the option to trigger a manual workflow from the applicable CRM Form.

It is possible to restrict, through use of the security model, which manual workflows can be triggered by a user.

Workflow fires asynchronously which means that there will be a noticeable delay between the workflow being triggered and it actually being processed. This in turn means that for any fields that are updated as part of a workflow rule, the changes will not be visible to a user until they reload the form after the workflow has been triggered, executed and completed.

It is also possible for workflow rules to reference external, custom, assemblies which have may be developed. These custom assemblies may be presented with dynamic input parameters and may also return output parameters.

Workflows only fire when the changes are made on the server which therefore means any changes that occur when a user is working offline will trigger the appropriate workflow rules when synchronisation occurs.

A power user who has undergone appropriate training should be able to generate and edit workflow rules.

An **example** of a workflow that may be configured:



When an Opportunity is marked as Won, the user's manager is notified by email.

<Or>

When a user runs a manual workflow rule, a case is routed to their manager.

1.2.2 Workflow Assemblies

Workflow assemblies are custom code that are packaged up and are made accessible from a workflow rule.

The key advantage of a workflow assembly is that because it is triggered through a workflow, a power user can choose when the code will be triggered and may also change the information that is passed to the workflow assembly.

An **example** of a workflow assembly that may be configured:

Share a case with the team that a user belongs to so that all members of the team can see the case.

1.2.3 Plug-ins

Plug-ins represent server side code that is triggered automatically as a result of programmatic action being taken on a specific record (i.e. when an account is created or updated).

Plug-ins are generally programmed to fire synchronously although there are options for them to be triggered asynchronously. When fired synchronously a user will not be able to continue to use CRM after triggering a plug-in until it has been fully executed and has completed its actions. This means that a user will typically see the results of a plug-in (i.e. the generation of an auto reference number) in real time but it also can mean that an error in the plug-in may prevent the user from completing an action such as saving a record.

Plug-ins may be developed to work when a user is offline although typically this is not the case which therefore means that they are normally only triggered when a change is made on the server. For a user who is working offline, the plug-ins will be fired as part of the synchronisation process.

A system administrator with appropriate training may be able to enable/disable plug-ins but it is not envisaged that they would be able to change the information that is passed to the code, nor would they be able to handle any information that is passed back from the plug-in.

An **example** of a plug-in that may be configured:

Generate a meaningful reference number and assign it to the contact record when a new contact is added to the system.



About Concentrix

Founded in 1999, Concentrix is a leading UK independent CRM specialist, providing quality business solutions to companies throughout the UK.

Concentrix implements [CRM software](#), systems and business solutions that are right for their customers' business requirements: matching their needs, goals and budget. Concentrix works across a wide variety of industry sectors with clients ranging from small companies with a handful of CRM system users, right up to household-name PLCs with hundreds of users on multiple sites.

Concentrix takes a 'product agnostic' approach to delivering CRM solutions to their clients. So, as well as Microsoft Dynamics CRM (which included [hosted Microsoft CRM](#) as well as on-premise deployment) Concentrix offers a range of other CRM software solutions including [Sage CRM](#), Sage [SalesLogix](#), and FrontRange's [GoldMine](#).

Concentrix is a Microsoft Gold Certified Partner, a fully Accredited Sage Business Partner and a FrontRange Premier Partner.

Concentrix is based in Mountsorrel, Leicestershire and has a satellite offices in Bristol and in central London. For further information about the information provided within this document, or any other products or services provided by Concentrix, please [contact Concentrix](#).

Concentrix Limited

20 Granite Way
Mountsorrel
Loughborough
LE12 7TZ

E: info@concentrix.co.uk
T: +44 (0)1509 410500
F: +44 (0)1509 410501

www.concentrix.co.uk

Registered in England No. 3891450
VAT Registration No. 738 0929 09