

# How to Write an IBM Rational Synergy™ Plugin

*Prepared for IBM Rational Synergy 7.1.0.1 iFix 03 or later*

Author: Troy Dugger  
Author: Alec Bray  
Author: Ranganath Prasannakumar

May 29, 2012

**IBM**

IBM Corporation, 1 New Orchard Road, Armonk, New York 10504-1722 United States

Document version: 1.0.0 19 May 2010

**License Agreement**

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of the copyright owner, IBM. Copyright infringement is a serious matter under the United States and foreign Copyright Laws.

IBM reserves the right to revise this publication and to make changes from time to time without obligation to notify authorized users of such changes. Consult IBM to determine whether any such changes have been made. The terms and conditions governing the licensing of IBM software consist solely of those set forth in the written contracts between IBM and its customers.

The information in this "How to ..." manual is subject to change without notice, and IBM does not assume any responsibility for any errors which may appear herein. No warranties, either expressed or implied, are made regarding the software described herein and its fitness for any particular purpose.

**Restricted Rights Legend/Notice**

Use, duplication or disclosure by the government is subject to the Restricted Rights restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or FAR 52.227-14, Alternate III, subparagraph (g)(3)(i), each clause of current date.

**Fictitious Names**

Any persons, businesses, government entities, academic institutions, locales or events used in this book and any associated presentation material are fictional and for the purpose of illustrating concepts related to the subject matter of this "How to..." manual. Any resemblance to actual persons (living or dead), businesses, government entities, academic institutions, locales or events is entirely coincidental.

**Trademarks**

IBM, the IBM logo and ibm.com are registered trademarks of International Business Machines Corporation in the United States, other countries, or both. IBM Rational Synergy is a trademark of IBM Corporation in the United States, other countries, or both.

OMG marks and logos are trademarks or registered trademarks, service marks and/or certification marks of Object Management Group.

Adobe, the Adobe logo, Acrobat, the Acrobat logo, FrameMaker, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Microsoft, Windows, Windows NT, Windows 2000, Windows 2003, Windows XP, the Windows logo and/or other Microsoft products referenced herein are trademarks of Microsoft Corporation in the United States, other countries, or both.

Sun, Sun Microsystems, Solaris, Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

All other product, service, or company names mentioned herein may be trademarks, registered trademarks, or server marks of their respective owners.

Licensed Materials - Property of IBM © Copyright IBM Corporation 2010. All Rights Reserved.

U.S. Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.



<b>1 ABOUT THIS HOW TO... GUIDE .....</b>	<b>4</b>
1.1 INTRODUCTION TO IBM RATIONAL SYNERGY PLUGIN ARCHITECTURE .....	4
1.2 BACKGROUND .....	4
1.3 OUTLINE ARCHITECTURE .....	5
1.4 ABOUT THIS HOW TO ... GUIDE .....	5
1.1.1 Description .....	5
1.1.2 Intended Audience .....	6
1.4.1 Aims and objectives of the Guide:.....	7
1.5 SUMMARY .....	7
1.6 TERMS .....	7
1.7 TYPOGRAPHY USED IN THIS GUIDE .....	8
<b>2 HOW TO WRITE A PLUGIN FOR SYNERGY 7.1.0.1 .....</b>	<b>9</b>
2.1 FILES YOU NEED .....	9
2.2 DEFINE A PLUG IN .....	9
2.2.1 Action .....	10
2.3 CREATING THE PLUG IN DIRECTORY (FOLDER) .....	11
2.3.1 Create plugin directory .....	11
2.4 WRITE THE PLUGIN.XML.....	11
2.4.1 Element Hierarchy.....	12
2.4.2 Keywords.....	22
2.4.3 Selecting Multiple Objects .....	24
2.4.4 Debugging .....	24
2.4.5 Outline plugin.xml file .....	25
2.5 WRITE THE *.PROPERTIES FILE .....	27
2.5.1 Outline properties file.....	28
2.6 LIMITATIONS AND ISSUES.....	28
2.7 WRITING A PLUGIN TO ACCESS SYNERGY COMMAND LINE COMMANDS .....	30
2.7.1 Session Identification .....	30
2.7.2 Running Synergy commands.....	30
2.7.3 Information returned from Synergy commands.....	31
2.7.4 Standard Out/Standard Error .....	31
2.8 FAULT FINDER.....	32
2.8.1 Menu does not appear in Menu bar or context menu.....	32
2.8.2 Menu appears but application does not start.....	32
2.8.3 Menu appears: Error message in Message View or log on clicking menu item .....	33
<b>3 SAMPLE PLUGIN .....</b>	<b>34</b>
3.1 UNIX/LINUX.....	34
3.2 WINDOWS .....	35
<b>APPENDIX: TYPE FINDER PLUGIN.....</b>	<b>36</b>
<b>NOTES AND WARNINGS.....</b>	<b>41</b>

# 1 About this How to... Guide

## 1.1 Introduction to IBM Rational Synergy Plugin Architecture

### Note:

**While IBM Rational Customer Support supports the Synergy plug-in technology to allow you to extend the user interface, it will not provide support for the scripts themselves. If you need assistance designing, writing, debugging or testing your custom scripts, contact your account manager about hiring IBM Software Services.**

**This technology is available in [IBM Rational Synergy 7.1.0.1 iFix 03](#) or later releases.**

The IBM® Rational® Synergy™ Software Configuration Management tool (known as IBM Rational Synergy) is a powerful, industry-strength tool with which to perform Software Configuration Management (SCM) for the development and delivery of software and other soft-copy artifacts. SCM is a *team discipline* for the construction of deliverable configurations, based on good practices for change identification, version control, reporting and audit. These practices provide an Enterprise with an audit trail of great use in corporate governance, providing records of *Accountability, Transparency, Safety, Cooperation, and Communication*.

IBM Rational Synergy's approach to SCM is concerned with making the process of producing software as efficient and as human-friendly as possible. The goal is to produce high quality software, release on release, to meet increasingly tight deadlines within budget.

To help satisfy the concerns and achieve the goal, [release 7.1 of IBM Rational Synergy](#) provides functionality to allow users to invoke scripts, applications or other utilities associated with Software Configuration Management directly from menu items (menu options) added into the menu structures provided in the tool. These additional menu options are provided as "plugins" to the menu systems: these plugins allow you to

- Create capabilities to extend the default functionality of the tool
- Support specific actions for your deployment
- Provide supplementary information on selected objects
- Run supporting scripts and applications asynchronously

Many of these activities have been available from the IBM Rational Synergy command line, of course: the advantage of the plugin architecture is that users do not have to set up a separate command line session to run specific queries or run specific actions.

**Note** that this plugin functionality is available for Synergy clients running in WEB MODE only (for discussions on the Synergy modes of operation, please refer to the appropriate Administration Guide for Synergy).

## 1.2 Background

[IBM Rational Synergy 7.1](#) included undocumented functionality to offer to IBM Rational Software Professional Services the ability to add menus and menu items to the Synergy client GUI interface. This feature had some limitations: in particular it required the user to edit and recompile Java code.

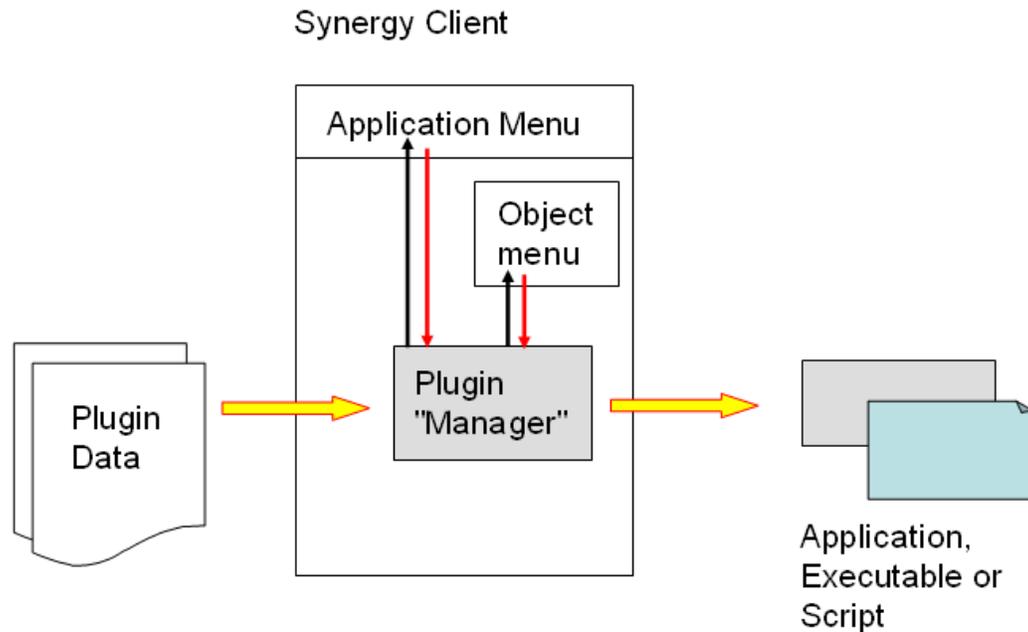
This functionality was, however, used to provide an integration with IBM Rational Harmony (all variants) so that users could access the correct tool mentors directly from the Synergy GUI. The function-

ality was also used to provide a “Database Documenter” feature, useful for enterprises concerned about Disaster Recovery.

These utilities proved the benefits of providing such a feature, and in [IBM Rational Synergy release and patch 7.1.0.1 iFix 03](#), the plugin functionality has been re-architected so that it is easier for consultants to provide these “application extensions”. At the same time, the re-architecture has made the feature more powerful and provides a platform for further development work.

### 1.3 Outline Architecture

The following diagram shows the plugin architecture in outline. The plugin itself does not provide any additional functionality, but provides the menu options and associated directives that allow a suitable application, executable or script to be run from within the IBM Rational Synergy client GUI.



*Note* that the plugin architecture for IBM Rational Synergy is likely to be revised as new applications are developed to take advantage of this functionality. Please ensure that you have the right version of this document for the version of IBM Rational Synergy that you are using. This document is for [IBM Rational Synergy 7.1.0.1 iFix 03](#) or later.

*Note* too that the menu items invoke free-standing scripts and applications which may, in turn, invoke Synergy command line commands.

### 1.4 About this How to ... Guide

#### 1.1.1 Description

This “How to ...” Guide provides an overview of the [IBM Rational Synergy Release 7.1.0.1](#) plugin functionality and shows you how to prepare a plug for use with IBM Rational Synergy.

This Guide has been designed to help you set up, configure and use IBM Rational Synergy 7.1.0.1’s plugin feature so that you can extend the set of features and functions offered from the Synergy GUI

interface: this allows you to further support software configuration management for your development and post-development teams within the software delivery chain.

To help you quickly get started with the IBM Rational Synergy plugin feature, we provide a simple example of each appropriate file in the text (following a full description of the options). These examples work with a Java application (supplied as part of a zip file) to return the Synergy type definition associated with a specific Windows file extension. The example plugin file and the application itself allow you to add a plugin to your Synergy GUI to see for yourself how it all works.

It is likely that any plugin that you provide will offer functionality that may, previously, been available only to Administrators or through the use of the IBM Rational Synergy Command Line interface. You may find, therefore, that the following manuals will also be of use to you as you construct your plugins:

[IBM Rational Synergy Administration Guide for UNIX](#)

[IBM Rational Synergy Administration Guide for UNIX \(Oracle\)](#)

[IBM Rational Synergy Administration Guide for Windows](#)

[IBM Rational Synergy Command Line Help](#)

[IBM Rational Synergy Classic Command Line Help](#)

**NOTE:** Links to all IBM Rational Synergy 7.1 documentation included Japanese translations can be found in the [Information Center](#).

### 1.1.2 Intended Audience

It is expected that plugins for the IBM Rational Synergy clients are likely to be provided by the “Tool Support Group”, Configuration Management team, or Synergy Administrators within an enterprise, although it is quite possible for individual users to construct and use their own plugins with no impact on any other users. A Team Leader or Build Manager, for example, might also wish to add a specific plugin for use by their development team.

In many cases, it is likely that the plugin architecture is used to bring together – as part of the Synergy client interface – scripts and applications that had previously been “scattered” in various “operational” directories (or folders) across the system: this ensures better control and execution of these scripts and applications (for one reason, the command line output is directed into the Synergy client log, tying Synergy operations and script operations together to provide a much stronger audit trail of activities).

Further, in situations where the operation of the servers themselves has been outsourced, running these scripts and applications needed to support Synergy can be brought back under the control of those users for whom the results are important and which are needed in a timely manner.

We do expect some basic working knowledge before you use this Guide. The first, and most important, assumption that we make is that you understand the reasons for the use of the IBM Rational Synergy Configuration Management tool within your enterprise. This helps guide you into the selection of functionality for use in the plugins.

You will have to have some knowledge of XML scripts.

#### 1.4.1 Aims and objectives of the Guide

Using this Guide you should be able to:

- Setup and configure a suitable plugin for the IBM Rational Synergy tool

## 1.5 Summary

The plugin architecture provided in [IBM Rational Synergy release 7.1.0.1](#) (and subsequent iFixes) allows users to add their own menu item to the IBM Rational Synergy menu bar or individual object popup menus. Using the top-level menu, users can then construct a menu hierarchy, with each menu item invoking a specific script or application. This “How to...” Guide describes how to do all this, and identifies some of the limitations of the current plugin architecture.

## 1.6 Terms

### Command Line Interface

A Command Line Interface (CLI) is, in general, a text-based method to allow users to interact with a computer operating system or with applications running on that operating system. This text-only interface contrasts with the use of a Graphical User Interface which typically uses multiple application-based windows, pictures (icons) to illustrate alternatives, menus to select options, and a pointing device (WIMP).

IBM Rational Synergy provides a complete suite of commands which are invoked directly through a Command line Interface.

### plugin

A plugin (there are other names) is a small software “element” that allows the capabilities of its invoking program to be extended.

The plugin architecture allows users to have the functionality they need without having to download an application that covers all possible features (“bloatware”).

It also allows users to add very specific functionality to their application installation.

### popup menu

This is also known as a context menu, a contextual menu, a shortcut menu or a pop-up menu: there is no one single standard name for this. A popup menu is a menu in a graphical user interface that appears when the user takes some action, typically a *right* mouse click operation (so this is also sometimes known as a right-click menu). A popup menu offers a limited set of choices that are available in the current state, or context, of the operating system or application. Usually the available choices are actions related to the selected object.

## 1.7 Typography used in this guide

Courier New

is used to show pathnames and filenames. It is also used to show command line commands

Arial

is used for window and dialog box titles, and dialog box fields, where we report these names directly in the text. ***Arial Bold Italic*** is used for buttons shown in windows or dialog boxes, when we report the buttons in the text.

## 2 How to write a plugin for Synergy 7.1.0.1

*Note:*

**For use with [IBM Rational Synergy 7.1.0.1 iFix 03](#) or later releases only.**

### 2.1 Files you need

For each plugin menu hierarchy you need a `plugin.xml` file and an associated `[plugin name].properties` file.

For each menu item in the menu hierarchy, you need a specific script, application or executable, or a script, application or executable the behavior of which can be modified by appropriate command line parameters.

### 2.2 Define a Plug in

To define a plugin for Synergy 7.1.0.1, you only need two XML files:

- `plugin.xml`
- `[plugin name].properties`

The `plugin.xml` is the menu definition file (and it must be named `plugin.xml`). This identifies the top level menu item and all the sub menu items, together with the actions to be taken when the user selects the menu item, when that menu item should be enabled and the list of any arguments to be passed to the action.

The `[plugin name].properties` file is a resource file that provides localized labels for the plugin's menu and menu items.

Both these files are placed in a directory under an `etc/plugins/[PLUGIN NAME]` directory (WINDOWS: `etc\plugins\[PLUGIN NAME]` folder) in the IBM Rational Synergy installation directory (`$CCM_HOME` or `%CCM_HOME%`). Each plugin must be placed in its own directory.

You can have as many plugins as you need.

*Note* that you are not able to add items to any of the default IBM Rational Synergy menus (Task, Edit, Find etc).

If you are creating an application-wide menu, the plugin menu is added to the IBM Rational Synergy client's Main window menu bar between the **TOOLS** and **Window** menu. If there is more than one plugin, then the plugin menus appear in alphabetical order based on the directory names in the `plugin` directory (folder).

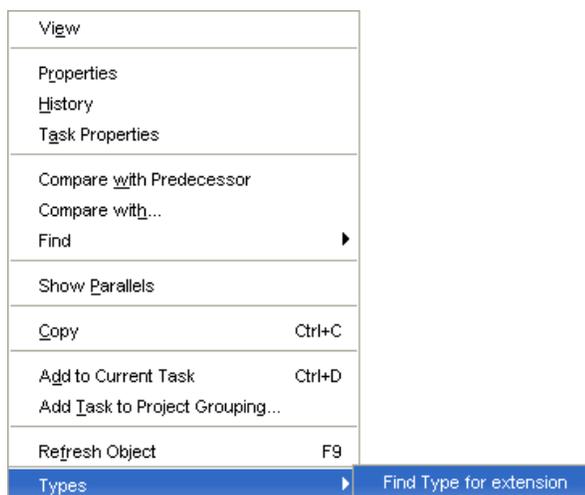


(*Note* that the plugin directory name need not match the plugin name itself: you can choose the directory name to affect the sequencing of the plugin menus in the Main application menu bar). In the

screen capture shown above, there are two plugins (the highlighted plugin menu is the subject of the example material in this Guide.)

If you want to add the menu and its submenus to the popup menus (also known as the context menus or shortcut menus) for the individual objects (and object versions) shown in the IBM Rational Client GUI, the new menu is added as the last item in the popup menu:

The illustration below shows the example (demonstration) plugin displaying on an object's popup menu:



If there is more than one plugin, then the plugin menus appear in alphabetical order based on the directory names in the `plugin` directory (folder).

*Note* that you can set conditions so that the popup menu item (context menu item, shortcut menu item) is displayed only for certain types of objects or for objects whose attributes have specific values. This means that you can tailor the plugin popup functionality very specifically to your needs. As an example, you can create the plugin so that one menu item is only displayed for graphics objects in an `integrate` state.

## 2.2.1 Action

You need an action script, executable or application for every sub-menu item in your plugin (except for any separators), or a single script, executable or application which can take command line arguments to identify which menu item has been selected by the user. A typical example of this latter approach is where a Java-based application is provided: the application provides a common framework with the command line arguments identifying which features or functions should be activated to support the menu choice.

In principle, you can invoke any script, executable or application from the plugin. In practice, there are some limitations, part due to the IBM Rational Synergy plugin architecture, part due to the scripting language environment itself.

Of the scripting languages used to date, the following potential issues should be noted.

`perl` (in a Windows environment):

---

`perl` is invoked in a UTF-8 environment, as the XML encoding is `<?xml version="1.0" encoding="UTF-8"?>`, but in this environment, without modification, `perl` does not accept command line arguments.

All scripts (and applications):

The script, executable or application's "Standard Output" (terminal or console output) (`stdout`) is written to the Synergy Client Message View (and to the logged-in user's `ccm_client.log`).

The script, executable or application's "Standard Error" output (`stderr`) may or may not be visible in the Message View (this depends on the error handling in the scripts and applications provided). For example, if a Java application is written to support the menu choices, the Java application should include stream handling to allow `stderr` to be written to the Synergy Client Message View.

## 2.3 Creating the plug in directory (folder)

*Note* that, to illustrate how to write a simple plugin, the example scripts shown in this Guide invoke a Java application to identify which Synergy type definition is associated with a specific File extension. The actual source is shown after each general description under the heading: `example`.

### 2.3.1 Create plugin directory

Each plugin has its own directory in the IBM Rational Synergy installation area `etc/plugins` directory (WINDOWS: `etc\plugins` folder).

*Note* that if this is the first plugin that you have provided for the specific client, you need to create the `etc/plugins` directory (WINDOWS: `etc\plugins` folder).

Unless the Synergy commands that you are intending to run require you to be running on the IBM Rational Synergy application server or the Informix Dynamic Server (where Informix is used as the underlying database), the plugins can be installed on any Synergy client machine. You may want to create an installer to add the plugin to all Synergy client machines that are to use the plugin.

#### **example**

Create directory:

```
$CCM_HOME%/etc/plugins/Types
```

```
%CCM_HOME%\etc\plugins\Types (Windows)
```

## 2.4 Write the `plugin.xml`

The `plugin.xml` file defines a single menu that contains all the menu items needed for the specific plugin.

*Note* that the `plugin.xml` file (and its associated `properties` file) is read in by the IBM Rational Synergy client at startup, so if you make any changes to the `plugin.xml` file (or to its associated `*.properties` file), you have to restart the Synergy client. However, the scripts or executables invoked by the `plugin.xml` file are called at runtime, so any changes to the scripts or executables do not need you to restart the Synergy client.

## 2.4.1 Element Hierarchy

This is the element hierarchy (structure) for the plugin XML file:

```
<xml header>
plugin
  compatibleversions
  version
  propertiesfilename
  menu
  labelkey
  item (optional)
    labelkey
    action
      enable_when (optional)
      exec
      arg (optional)
      function (optional)
      menu (optional)
  separator (optional)
```

### xml file header

Standard start line for XML files, and is included here for completeness. For more information please refer to the W3C recommendation “Extensible Markup Language (XML) 1.0 (Third Edition)”.

The version of XML and the encoding used. These are required elements.

xml header			
Attributes	version	1.0	Required
	encoding	UTF-8	Required

```
<?xml version="1.0" encoding="UTF-8"?>
```

### plugin

This is the root element.

The `plugin` element can have two attributes, both of which are optional. It has three subelements, one of which is optional.

plugin			
Attributes	version	The version of the plugin itself (configuration management information)	Optional
	name	The name of the plugin	Optional
Subelements	compatibleversions		1
	propertiesfilename		1
	menu		1

### compatibleversions

The `compatibleversions` element identifies those releases of IBM Rational Synergy with which the plugin has been tested and which are known to support the plugin.

The `compatibleversions` element supports one or more `version` subelements.

compatibleversions		
Subelement	version	1 or more

### version

The `version` element is required and needs to match the Version string used to define the IBM Rational Synergy release. You can find this string by opening a suitable command prompt (Windows) or a terminal session (on UNIX) and entering:

```
ccm version
```

The value that you need to use is the first line returned from the `ccm version` command:

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\demo>ccm version
IBM Rational Synergy Version 7.1
IBM Rational Synergy Patch Version
7.1.0.1 This is Patch 7.1.0.1 for IBM Rational Synergy Release 7.1
7.1.0.1.03 IBM Rational Synergy Fix patch 7.1.0.1 03 for Release 7.1

C:\Documents and Settings\demo>

```

The `version` element does not contain any subelements.

version		
Value	The IBM Rational Synergy version which supports this version of the plugin.	Required

*Note* that the version string needed for IBM Rational Synergy release 7.1.0.1.03 is “**7.1**”.

**Note**, too, that there is no implicit inclusion in the `version` string. For instance, if you specify **7.1** as a compatible version, then this does *not* mean the plugin can run on Synergy version 7.2 (or on Synergy version 7.0). You must list *each* Synergy release (using the major and minor release numbers only) to define on which Synergy version the plugin has been tested and for which it has been found to work correctly.

### propertiesfilename

The `propertiesfilename` element is a required element.

The value of this element defines the name of the file that contains the localized strings for each `labelkey` element value defined in the `plugin.xml` file.

The name of the properties file can be any suitable name, but we recommend that, where appropriate, you use the following guidelines:

1. The properties file should use the extension  
\*.properties or  
\*.xml\_properties
2. Localized properties files should each use the same filename but with a locale specific suffix:  
for example

```
[my_plugin]_ja.xml_properties
[my_plugin]_fr_CA.xml_properties
```

In this case, the locale identifiers are defined by this format: `[language[_territory][.codeset][@modifier]]`.

The `propertiesfilename` element does not contain any subelements.

propertiesfilename		
Value	The name of the properties file which contains the localized strings for the menu labels..	Required

*Note* that the properties file is located in the same directory (folder) as the `plugin.xml` file which uses it. In this case, there is no path information needed.

*Note* also that multiple plugins can each use a similarly –named properties file (as per the suggested naming convention above) as each properties file resides in its own plugin directory (folder).

**menu**

The `menu` element is required. It defines the menu labels and the associated actions for the plugin.

The `menu` element contains one `labelkey` sub element which in turn defines the menu name (the root of this menu hierarchy).

Within this menu there can be multiple menu items and multiple submenus.

**labelkey**

Each `menu` element and each `item` element requires its own `labelkey` element.

The `labelkey` value itself can contain any type of characters. The value is used to find the localized string in the appropriate localized properties file which is then used in the display for the associated menu or item element.

The `labelkey` element does not contain any subelements.

labelkey		
Value	Used to find the localized display string to be used as the label for the menu or (menu) item from associated properties file.	Required

**item**

The `item` element defines a menu item within the plugin menu hierarchy.

You can associate any number of actions for each menu item.

item		
Subelements	labelkey	1
	action	1 or more

*Note* that you must associate a `labelkey` (and associated label) with a specific `action`. A label without an action acts as a separator, but there is a specific `separator` element that you should use in this case.

**action**

The `action` element defines (by means of the `exec` and `arg` elements) the action that IBM Rational Synergy has to take when the user clicks on the menu item (if the user clicks on the root of the menu hierarchy, this displays the menu itself).

The action can be to open a URL or to run a defined script, application or executable.

You can provide as many actions as needed to support the menu item. If there is more than one `action` element, however, each `action` must have the `platform` attribute defined. The valid platform values are 'Windows' and 'UNIX' (case-insensitive) indicating which file or URL applies to which

platform. The `platform` attribute indicates which file to execute or URL to show, depending on the OS that is running the IBM Rational Synergy client.

The `action` has to have one `exec` subelement, but there can be as many `arg` elements as may be needed to support the command line arguments needed to be handed across to the operation defined in the `exec` subelement.

Since Synergy attributes can contain special characters (characters like #, &, / etc.) which might be misinterpreted by a web browser when they are substituted for keywords in a URL by the plugin framework, an optional `urlencode` attribute is provided to make sure the user can choose to encode the query string part of the URL (the part after the ? character) using the UTF-8 encoding scheme. The value of this attribute can be anything since the plugin framework only checks for the presence of the attribute and performs url encoding as already explained. Please be aware that this attribute cannot be specified without a value since strict XML parsing does not allow the presence of valueless attributes. Encoding is done only if the action string has a ? character and anything following it will be encoded.

For example, if the `urlencode` attribute is absent and the `exec` tag for the action looks like `<exec>http://server.com/report.php?SELOBJ=%context_project:objectname%</exec>`

Lets say the value substituted for the keyword `%context_project:objectname%` is `"bb_intr_api~tut71#ta_current_20110830:project:kok_aud#1"` for a reason that version delimiter on the Synergy database is '#' and the DCM delimiter is also '#'.

In the above case, the web browser will strip the second '#' character since only one such character is allowed as part of any URL without proper encoding and this character has a special meaning of being the reference anchor name indicator within the page.

The value will be `"bb_intr_api~tut71%23ta_current_20110830%3Aproject%3Akok_aud%231"` (notice the special characters being replaced by a number from the url encoding scheme) if the `urlencode` attribute is present in the corresponding action tag (*which may look like `<action urlencode="1">`*) and the final URL will be handled properly by any web browser.

*Note: It is recommended that url encoding be used only if the plugin action launches a URL in a web browser.*

Action			
Attributes	<code>urlencode</code> (only 7.1.0.6, 7.2.0.3 or later)	The presence of this attribute indicates that the query string part of the generated URL should be urlencoded: Any	Optional
	<code>platform</code>	The Operating System needed to run this specific action: Windows <i>or</i> UNIX	Required if more than one action
Subelements	<code>enable_when</code>		0 or 1
	<code>exec</code>		1
	<code>function</code>		0 or 1
	<code>arg</code>		0, 1 or more

## enable\_when

The `enable_when` subelement allows the popup menu (context menu, shortcut menu) to be displayed only if all the conditions identified in the subelement value are matched.

If the `enable_when` subelement is not present, the menu appears only on the Synergy Client application main menu bar.

**Note** that the termination for the `enable_when` tag is `/>` (forward-slash right-chevron) (the right-chevron is, of course, the standard XML tag termination character).

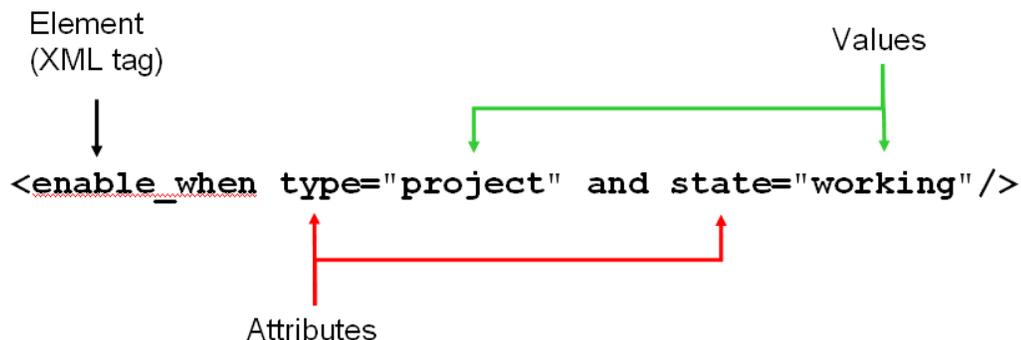
The `enable_when` subelement has a mandatory attribute: `type`. You must specify the Synergy object type(s) for which you want the popup menu item to display. If the `enable_when` subelement is present and the `type` is set to the wildcard value `type="*"`, the menu appears on the popup menu (context menu, shortcut menu) for every object.

By setting the `enable_when` subelement attributes, the specific menu item (and its action) (`labelkey` → `label` → `exec`) only appears, and is valid, for specific objects (or object versions) in the Synergy client GUI - wherever the object is shown in the GUI: this may be in the Project Pane, for example, or in the History View, Query returns or Properties dialogs or other places where the object or object version is displayed.

The set of default values for the `type` attribute provides for standard Synergy object types associated either with project structure or with processes and process (update) rules. However, the operation of the `enable_when` subelement depends, to a considerable extent, on the way that you have chosen to define your own Synergy types for the objects that are used in the Synergy projects (source files, for example)(*note* that these are not the same as the Microsoft Windows file associations). The IBM Rational Synergy type definitions define how an object (and all its associated object versions) is handled within the tool. There are two supertypes, `ascii` and `binary`, but within these branches, Synergy Administrators can create their own types and, indeed, hierarchies of types (Synergy comes supplied with a useful starting set of object types).

The larger the number of object types that you have set up in Synergy, the more control you may have over the display of the popup menu (context menu).

As an example, the menu item that includes this `enable_when` element only appears, and is valid, for `working` state projects:



*Note* that the `enable_when` subelement includes attributes which map directly to IBM Rational Synergy attribute names (as available through the `ccm attr` command line command), but not all Synergy attributes are mapped to the `enable_when` subelement attributes.

However, for those Synergy attributes which are not “natively” addressed in the `enable_when` subelement, you access the attribute using “attribute” and “value” attribute pairs. This allows the `enable_when` subelement to access any attribute on the specific Synergy object, including any custom (user-defined) attributes that you have provided.

*Note* that there is a wildcard option, `type="*"`, which means that the popup menu (context menu) is displayed for any selected object, regardless of type.

The `enable_when` subelement provides a single XML tag to identify when the menu and submenus should be displayed for a specific object: this makes it easy to construct specific menu displays.

enable when				
Attributes	name	permitted values		
	condition	equal notequal	This allows you to specify an action on negative conditions.	
	type	* Projects Project Groupings Tasks Files Directories Baselines Process Rules Releases Folders Folder Templates Processes Objects	All objects  The value can also be any one of the actual IBM Rational Synergy types (as in ccm query "cctype='*'")	
	state	working integrate prep released rejected	The value can be any valid Synergy status. Multiple statuses can be used, separated by the delimiter (which see)	
	attribute	Any valid Synergy attribute name	The attribute and value work in pairs to define the objects' attribute and value to use to define if this menu should be enabled for the object. Multiple values can be used, separated by the delimiter (which see)	
	value	The value(s) of the attribute identified in the attribute attribute (above)		
	number_selections	1		default value
		n		n is any number
		n+		n or more
	user_privilege	One or more valid Synergy roles. Roles include: developer build_mgr component_developer		The value can be any valid Synergy role. Multiple roles can be used, separated by the delimiter (which see)
delimiter	,		The character used to separate multiple values for the attribute. The default value is comma ( , )	

By using multiple `enable_when` elements within a specific `action`, you can effectively create logical AND or logical OR conditions.

If the multiple `enable_when` elements have the same `type` attribute value, the multiple `enable_when` conditions are logically ANDed, as in this example:

```
<action>
  <enable_when type="task" state="task_assigned"/>
  <enable_when type="task" attribute="owner" value="gbalbr"/>
  <exec>notepad.exe</exec>
</action>
```

The plugin popup menu (context) menu is only displayed if there is a task assigned to user gbalbr.

If the multiple `enable_when` elements each have a different `type` attribute value, the multiple `enable_when` conditions are logically ORed, as in this example:

```
<action>
  <enable_when type="project" state="working,prep"/>
  <enable_when type="java" attribute="owner" value="gbalbr"/>
  <exec>notepad.exe</exec>
</action>
```

In this case, the plugin popup menu (context) menu is displayed if the project is in a `working` or `prep` state, *or* the object is a `java` file owned by user gbalbr.

### 2.4.1.1 Restrictions

You need to be aware of some limitations or restrictions when you use the element `enable_when`. Most of these restrictions are concerned with the display of the menu item on the main menu and/or the selected object's popup menu (context menu).

1. You can not construct a menu where a menu item is only in the Synergy Client application main menu and is only enabled when there are specific objects selected in the Synergy Client.

This is unlikely to be a real problem. The "natural" place for users to look for a menu item for a selected object is the popup menu for the object. There are examples of applications which only offer main application menu choices for selected items – both Lotus Notes and Microsoft Outlook (to some minor extent) do this, for example – but it can be irritating for the user.

2. You can not construct a menu where a menu item is only in the selected object's popup menu (context menu) and *not* in the Synergy Client application main menu. The menu item always appears in the Synergy Client application main menu.

There are two "schools of thought" on this. If the menu item refers solely to an individually selected object, the menu should not necessarily appear in the Application main menu bar. This minimizes the chance that an object can be selected and a main menu item is then (accidentally) selected – to action something that then operates on the selected object.

However, where a menu item refers to a set or selection of objects, it may well be legitimate to put the menu item in the Application menu bar. In any case, the UNIX MOTIF style guide implies that a menu item should be present at more than one location: it goes on to state that the popup menu should be a redundant element of a user interface whose main purpose is to provide the user with quick access to

common tasks for individual objects. Following this style guide, it is appropriate to have a menu item in the main application menu bar and in the selected object's popup menu (context menu).

Generally, having the menu item in two locations is not likely to prove to be a restriction.

3. You can not construct a menu where a menu item is shown on the popup menu (context menu) if there are no selections.

However, if the end user *right-clicks* on an object (version) in the Synergy Client Project Pane (or any other dialog or show screen in the IBM Rational Synergy GUI) to show the popup menu (context menu), this object (version) is necessarily selected.

## exec

The `exec` element defines the actual command to be executed or the URL to be referenced.

exec		
Value	A valid URL <i>or</i> Path to a file (the path may be a full or relative path) (you may optionally include the command line arguments in the <code>exec</code> value)	Required

If your Java Virtual Machine includes a suitable JSR 223 Script Engine for the file type defined in the path (that is, if the filename extension is one recognized and supported by the Script Engine), the file is then executed by the Java Virtual Machine (JVM). If there is no suitable Script Engine for the extension, then your file is executed as a separate process.

(For more information on JSR 223 Script Engines, please refer to the Java Development web site: <https://scripting.dev.java.net/>)

The `exec` element does not contain subelements (*note* that the `arg` subelement is a subelement of `action`)

## function

**Note** that the `function` subelement is optional and is *only* valid for JSR 223 supported languages.

function		
Value	Identifies the function in the invoked script that is to be called when the user clicks on the menu item	Required

You use the `enable_when` subelement to define when the menu option is allowed and enabled for any specific object. This means that the functions you provide (and identify in this subelement) do not need to include any filters or processing to determine the type of object selected.

## arg

The `arg` subelement is optional. It can be used instead of specifying all arguments in the `exec` element.

IBM Rational Synergy uses the `Java Runtime.exec(String[])` method to launch the script specified by the `exec` value. In the section describing how to use Synergy command line arguments as part of the plugin, the `arg` subelement(s) are used to force the use of the `Java Runtime.exec(String[])` method: this makes it easy to parse the return values.

The `arg` element is used to provide the command line arguments to the script or executable defined in the `exec` element. You can use as many `arg` subelements as you need to provide the set of command line arguments needed for the script or executable. Generally, using the `arg` subelement(s) are likely to make it easier for you to maintain the `plugin.xml` file.

If the script or executable is running IBM Rational Synergy command line commands or needs to work with certain IBM Rational Synergy data, the `arg` element can report the values for a number of Synergy keywords (see below).

The `arg` element does not contain any subelements.

arg		
Value	A command line argument <i>or</i> IBM Rational Synergy keyword	Optional

**Note** that if you provide `arg` elements but you are using a JSR 223 compatible script, the `arg` elements are not used: the JSR 223 compatible scripts can get information from the Java Virtual Machine (JVM), so these scripts do not need to be passed information explicitly.

## separator

The `separator` element defines a menu separator within the plugin menu: this allows you to partition the menu items to make the menu easier to read and to access for the users. You need to consider your menu design with care. There are many cases where separators can help the users: for example

- Separate commonly used menu items from those more rarely used
- Separate functional groupings into distinct areas

The `separator` element does not contain any subelements (or any values).

separator		

**Note** that the termination for the `separator` tag is `/>` (forward-slash right-chevron) (the right-chevron is the standard XML tag termination character).

You can use a separator anywhere in your menu. However, if you have two or more separators together, only one is displayed – this is for cosmetic reasons. Also, if you put a separator at the top or bottom of your menu, the separator is ignored.

## 2.4.2 Keywords

IBM Rational Synergy information can be passed directly to the script, application or executable defined by the `exec` element by using specific keywords. The keyword is placed in an appropriate `arg` element (keywords such as `%ccm_home%` which identify paths can also be used in the `exec` element). The value associated with the keyword is substituted at runtime.

**Note** that if you use a keyword which expands to a path (keywords such as `%ccm_home%`, `%database_path%` and `%plugin_path%`) and if there may be spaces in the path name, the complete path must be enclosed between double quotes (" . . ")

As an example: if you are running a Java application, and you are intending to use the java Runtime Environment (JRE) supplied with IBM Rational Synergy, you would use double quotes at the start and end of the `exec` element:

```
<exec>"%ccm_home%\jre\bin\java"</exec>
```

**Note** that the sequencing of the `arg` elements following the `exec` element needs to match the sequence that the script or application expects to receive the command line arguments:

Some of the keywords are more likely to be relevant if the menu is invoked from the IBM Rational Synergy Main menus bar (examples include `%ccm_home%`) whereas other keywords are more suitable for use in the object popup menu (context menu).

<b>keywords</b>	
<code>%ccm_home%</code>	The full path to the IBM Rational Synergy installation area.
<code>%context_directory:&lt;attributename&gt;%</code>	The value of the attribute identified by <code>&lt;attributename&gt;</code> of the Synergy project directory which contains the selected object.
<code>%context_directory:fullpath%</code>	The full path to the work area directory (folder) which contains the selected object
<code>%context_project:&lt;attributename&gt;%</code>	The value of the attribute identified by <code>&lt;attributename&gt;</code> of the project which contains the selected object As examples, <code>%context_project:wa_path%</code> identifies the work area path of the project which contains the selected object.: <code>%context_project:displayname%</code> provides the Synergy display name for the project which contains the selected object.
<code>%current_task:&lt;attributename&gt;%</code>	The value of the attribute identified by <code>&lt;attributename&gt;</code> for the current task. (substitute the attribute required for <code>&lt;attributename&gt;</code> ).
<code>%database_name%</code>	The name of the currently accessed Synergy database
<code>%database_path%</code>	The full path to the currently accessed Synergy database
<code>%plugin_path%</code>	The full path to the directory where the <code>plugin.xml</code> file is located.
<code>%selected_object:&lt;attributename&gt;%</code>	The value of the attribute identified by <code>&lt;attributename&gt;</code> of the selected object
<code>%selected_object:fullpath%</code>	The full path to the work area representation of the selected object

### 2.4.3 Selecting Multiple Objects

You can define a menu item to work on multiple selected objects, but you can only do this if you are writing a script or application in a JSR 223 compatible language: it does this by accessing the entire context information of the GUI.

If you are rewriting a script in a language which is not JSR 223 compatible, you can not apply the plugin script to multiple selected objects. However, in this case, you could perform an operation on the files in a selected directory (object) or selected project (object) (but *note* that this operation applies to *all* files in the selected higher level object).

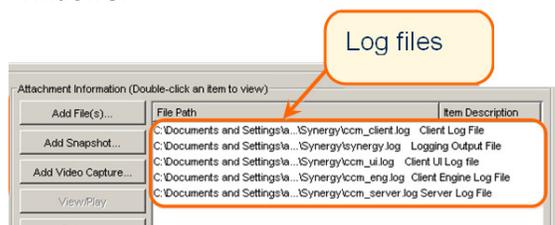
### 2.4.4 Debugging

You are advised to use a good XML editor, as this will alert you to any missing tag termination characters, for example.

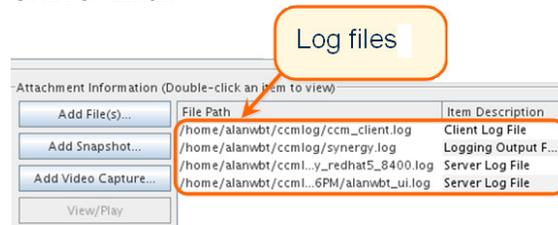
If the plugin does not work, check the `synergy.log` file. In a Windows client, the file is found in the user's area in the `..\Application Data\IBM Rational\Synergy` folder (the full paths are different for Windows 7 compared to the earlier versions of the Windows OS). In a UNIX or Linux client, the log file is located in the user's home directory.

If you are not sure how to access the `synergy.log` file, as described above, you can view this file through the Synergy GUI interface using the **Help** → **Submit Support Request** menu option. The Support Request finds the appropriate log files for the client and attaches these log files directly to the report. You can view the appropriate file (select the `synergy.log` file and then click **VIEW**) You then **CANCEL** the Support Request.

#### Windows:



#### UNIX or Linux



*Note* that the `synergy.log` file is the debugging logfile, not the Synergy messages log file (which lists the standard application error messages, notifications and so on).

Check the `synergy.log` for information that should help you understand why a plugin menu is not created or why an action is not added to the menu.

If you need more information about why an action enables or disables, you can get more debug information. Edit the IBM Rational Synergy installation area `etc\logging.properties` file (for Windows clients) (`etc/logging.properties` file for UNIX and Linux clients).

Add these lines to the end of the `logging.properties` file:

```
# For Synergy plugins, increase the logging level
com.telelogic.cm.ccm_client.actions.LaunchBrowserPluginAction.level = FINEST
```

*Note* that, even with this logging in place, under certain circumstances it can be difficult to identify why the plugin is not working.

If possible, you should desk-check your `plugin.xml` and its associated `properties` file - double-checking file names, for example – and review the material with a colleague before installing the plugin for the first time.

*Note* that you need to restart the IBM Rational Synergy client each time that you make changes to the `plugin.xml` file or to its associated `*.properties` file, as these are read in at start time. However, the scripts or executables invoked by the `plugin.xml` file are called at runtime, so any changes to the scripts or executables do not need you to restart the Synergy client.

## 2.4.5 Outline `plugin.xml` file

An outline structure for a simple `plugin.xml` file (as developed in the example) is:

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin version="[version]" name="[plugin name]">
  <compatibleversions>
    <!-- for SYNERGY 7.1 and 7.1 Fixes -->
    <version>7.1</version>
  </compatibleversions>
  <propertiesfilename>[name].properties</propertiesfilename>
  <menu>
    <labelkey>[MainWindowMenuName]</labelkey>
    <item>
      <!-- [description] -->
      <labelkey>[MenuItemLabel1]</labelkey>
      <action>
        <enable_when type="*" />
        <exec>[script name]</exec>
        <arg>[arg1]</arg>
        <arg>[arg2]</arg>
        <arg>...</arg>
        [ repeat arg structure for each
          command line argument ]
      </action>
    </item>
    <separator/>
    <item>
      ...
    </item>
    [ repeat item structure for each menu item ]
  </menu>
</plugin>
```

### example

*Note* that this example is prepared for use with an IBM Rational Synergy Windows Client.

This `plugin.xml` file is used to invoke the Type finder application.

The plugin provides `%ccm_home%` as an argument to the Java executable, as the executable has to invoke the IBM Rational Synergy `ccm` command (and to do this, uses the full path to the IBM Rational Synergy installation `bin` directory).

`%ccm_home%` is also used directly in the call to the Java Runtime Environment (`java`) supplied with the IBM Rational Synergy distribution: this helps to ensure that there are no compatibility issues. `%ccm_home%` is used both in the path to the JRE and as an argument to the Java application. (This is demonstration material!)

The application that is invoked is the Java Runtime Environment (`java`): the two command line arguments for this are `-jar` and the path to the Java archive file (`jar` file). The Java application itself expects a single command line argument, `%ccm_home%`: this is the third argument (in total) for the plugin.

*Note* that this example does *not* display on the individual object's popup menu (context menu, shortcut menu).

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin version="1.1" name="Synergy Types Assistant">
  <compatibleversions>
    <!-- for IBM Rational SYNERGY 7.1.0.1 iFIX 03 -->
    <version>7.1</version>
  </compatibleversions>
  <propertiesfilename>TypeAssist.properties</propertiesfilename>
  <menu>
    <labelkey>typeAssist</labelkey>
    <item>
      <!-- Plugin Demo: Find Type for Extension (WINDOWS) -->
      <labelkey>findType</labelkey>
      <action>
        <exec>"%ccm_home%\jre\bin\java"</exec>
        <arg>-jar</arg>
        <arg>"%plugin_path%\typeInfo.jar"</arg>
        <arg>"%ccm_home%"</arg>
      </action>
    </item>
  </menu>
</plugin>
```

## 2.5 Write the \*.properties file

This is the element hierarchy (structure) for the properties file:

```
<xml header>
<Document type>
properties
  entry
```

### xml file header

Standard start line for XML files, and is included here for completeness. For more information please refer to the W3C recommendation “Extensible Markup Language (XML) 1.0 (Third Edition)”.

The version of XML and the encoding used. These are required elements.

xml header			
Attributes	version	1.0	Required
	encoding	UTF-8	Required

```
<?xml version="1.0" encoding="UTF-8"?>
```

### Document type

Standard document type information for XML files, and is included here for completeness.

```
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd"
>
```

### properties

This is the root element.

The `properties` element has as many `entry` key subelements as needed to support the list of menu items. You need as many `entry` key elements in the `properties` file as you have `labelkey` elements in the `plugin.xml`.

properties		
Subelements	entry	1 or more

### entry

The `entry` element provides a localized display string to show in the IBM Rational Synergy GUI Main menu or popup menu (context menu, shortcut menu) for the associated `labelkey` value in the `plugin.xml` file.

There is one `entry` element for each `labelkey` element.

entry			
Attributes	key	labelkey Value	Required

The `entry` element's `key` value provides the “look up” key for required menu item name. The value provided in the `tag` itself is the localized string to be used for the menu or menu item.

One `entry` element provides the main menu item (the root of the plugin menu hierarchy): the other `entry` elements provide the submenu items.

You can have as many properties files as you need for any localized versions of the plugin, but only one properties file is invoked for a Synergy session.

## 2.5.1 Outline properties file

An outline structure for a simple `[name].xml_properties` file (as developed in the example) is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd"
>
<properties>
  <entry key="[MainWindowMenuName]">Menu Name</entry>
  <entry key="[MenuItemLabel1]">Menu Item One</entry>
  [ repeat entry key structure for each menu item ]
</properties>
```

### example

(The filename is: `TypeAssist.properties`)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd"
>
<properties>
  <entry key="typeAssist">Types</entry>
  <entry key="findType">Find Type for extension</entry>
</properties>
```

## 2.6 Limitations and Issues

There are some limitations with the plugin functionality provided with IBM Rational Synergy release and patch 7.1.0.1 iFix 03.

### perl (in a Windows environment)

`perl` is invoked in a UTF-8 environment, as the XML encoding is `<?xml version="1.0" encoding="UTF-8"?>`, but in this environment, without modification, `perl` does not accept command line arguments.

### Character encoding

If you are running non-JSR 223 compatible scripts, these are executed using the Java `ProcessBuilder` functionality. This can cause issues, as the character encoding is the Operating System's default character encoding.

If you are using an IBM Rational Synergy database that substitutes keywords for characters in one encoding system, but the Synergy client GUI is running on a different character-encoding platform, there is an issue. You will have to ensure that the plugin code itself can cope with these issues of character encoding (*note* that the simple demo used with this Guide does not deal with this issue of character encoding).

If the character encoding is a big issue, you are advised to redesign the plugin to use a JSR 223 compatible scripting language.

### Localization

Only one properties file is available for the session. If you are providing specific localized variants for the plugin, you need to be careful about when and how you switch the localized properties file.

### `%CCM_HOME%`

If you are invoking an MS-DOS batch file in the `<exec>` element of the `plugin.xml`, you need to be careful how you deal with the Synergy keyword `%ccm_home%`.

If you have already set the value of the environment variable `%CCM_HOME%` for the Windows PC, this is expected to be the same value as that provided by the IBM Rational Synergy keyword `%ccm_home%`

Please *note* that MS-DOS batch files invoked through the plugin menus do not have to set the `CCM_HOME` environment variable as the MS-DOS batch file is launched within a process that has this environment variable already set.

Generally you should not need to set the value of the `CCM_HOME` environment variable. However, you may need to do this if your batch file *either* is invoked through the plugin menu *or* is invoked by an external means, for example by a scheduler. If your batch file includes a line such as

```
set CCM_HOME=%1
```

and you are passing the argument `<arg>%ccm_home%/arg>` as the first command line command, your script uses the value passed by the plugin (this makes a local substitution of the value for the `CCM_HOME` environment variable).

This should not be a problem, but might be a source of confusion if you have more than one version of the IBM Rational Synergy client installed on your machine.

## 2.7 Writing a Plugin to access Synergy command line commands

There are two exciting uses for the IBM Rational Synergy plugin functionality:

- Extend the functionality of the tool
- Provide access to existing scripts and utilities from the Synergy GUI

In both cases, it is likely that you need to invoke Synergy command line commands.

*Note* that the plugin architecture is only available for Synergy clients running in WEB MODE, so you have to make sure that you are using the WEB MODE versions of the Synergy command line commands (where there is a difference between the current two modes of operation).

### 2.7.1 Session Identification

The IBM Rational Synergy plugin architecture provides an environment within the current running session in which to run the script, application or executable invokes by the plugin. Any Synergy command line command that you run uses the same session that is used by the Synergy client.

You do not need to set the `CCM_ADDR` environment variable in any script or executable that you supply – you automatically connect to the session that the Synergy client is using. However, you can not access this variable as it is not set.

Because you are using the same session as the client, you do not need to start or stop a Synergy session in order to run `ccm` commands: this makes it very easy to prepare scripts to analyze issues or extend functionality.

If you are porting any existing scripts to this plugin architecture, you are advised to strip out the commands to start, identify and stop a Synergy session. These commands are typically

```
[set] CCM_ADDR = `ccm start -nogui -q ....`  
ccm stop
```

In particular, make sure there is no `ccm stop` command in the script or executable – you will kill the client session from the plugin (unless, of course, your plugin is intended to take specific actions and then kill the session).

*Note:* In Synergy 7.2 and later releases, using Synergy plugin, user is not allowed to change the default role by using "ccm set role" command.

### 2.7.2 Running Synergy commands

Provided that the language you are using allows you to do this easily, it is a good idea to provide an option to allow the IBM Rational Synergy command that you are running to be reported, when appropriate, to the IBM Rational Synergy Message View.

There are a number of reasons for this:

1. It provides an audit trail of the Synergy commands which are run
2. It allows the Synergy command to be captured and run independently
3. It shows the progress of the plugin action

Whilst this is most useful during the set up phase of preparing a plugin, the auditability of the plugin is frequently useful in normal operating circumstances.

Here is an example from a `ccm_client.log` file:

```
03/10 09:24:15 Executing Synergy Command: D:\IBMRational\Synergy7.1\bin\ccm delimiter
```

### 2.7.3 Information returned from Synergy commands

You need to consider what you are going to do with the information returned from any Synergy commands. If you have been running Synergy command line commands from an MD-DOS command prompt or from a terminal session, you may have either observed the information as it is displayed or you may have redirected the output to a file. While this latter option is still viable, you may need to consider more appropriate ways of handling the output.

If you are running the Synergy commands from an MS-DOS batch file, your best option is to write the Synergy command return information into a suitable file. If your command is expecting to return one single entry, you can use the `FOR` syntax, as in this example:

```
for /F %i in ('ccm delimiter')do set CCM_DLIM=%i
```

However, you are more likely to use a language such as `perl` or `Java` for your plugin – the results from the Synergy command can be stored in a `perl` array or `Java` String list (or, for any language, any structure which is dynamically resizable). This example is taken from `Java`:

```
public static List<String> slCommandReturns = new Ar-  
rayList<String>();
```

The data in the array or String list (or other structure) is now available for analysis.

### 2.7.4 Standard Out/Standard Error

Any items which would normally be displayed as “Standard output” (`stdout`) are displayed in the **IBM Rational Synergy Message View** (and are written to the current user’s `ccm_client.log` file).

You can send messages to the **Message View** and `ccm_client.log` file simply by using the `print` function of the language that you are using: you do not need to write to a separate log file (unless you wish to do so).

Here is an example of a plugin reporting into the Synergy `ccm_client.log` file:

```
04/13 05:41:31 User demo on database \\tls2009\ccmdb\harmonySE  
using server http://tls2009:8400  
04/13 05:46:12 IBM Rational Synergy plugin: Type Finder  
04/13 05:46:12 Searching for Synergy type for extension : .py  
04/13 05:46:24 No Synergy type matches the given extension.  
04/13 05:50:27 Quit
```

You might need to make sure, if you are using stream-based languages, that the Standard Output stream is actually going to where you expect.

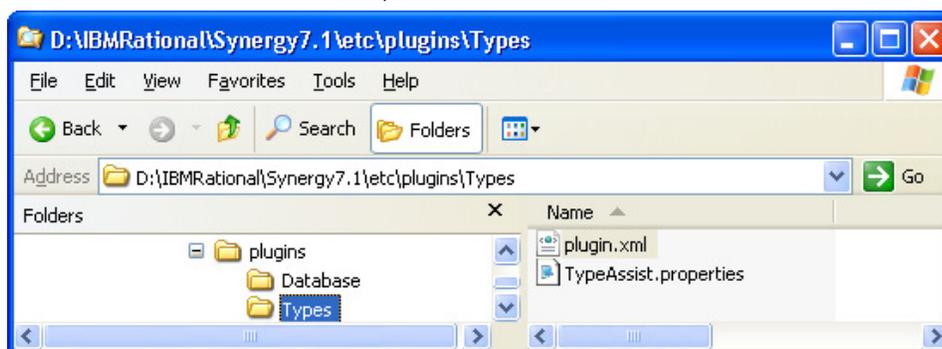
Standard Error (`stderr`) is *not*, by default, written to the IBM Rational Synergy Message View. This means that you may miss error messages which are generated by the interpreter or runtime environment that you are using. If you are using a stream-based language, you are advised to prepare a “stream gobbler” or stream capture function (class, procedure, etc) to collect the error output and redirect this to the Synergy Message View.

## 2.8 Fault Finder

### 2.8.1 Menu does not appear in Menu bar or context menu

Check that the properties file has the correct name as set in the `plugin.xml`'s `propertiesfilename` element:

```
<propertiesfilename>TypeAssist.properties</propertiesfilename>
```

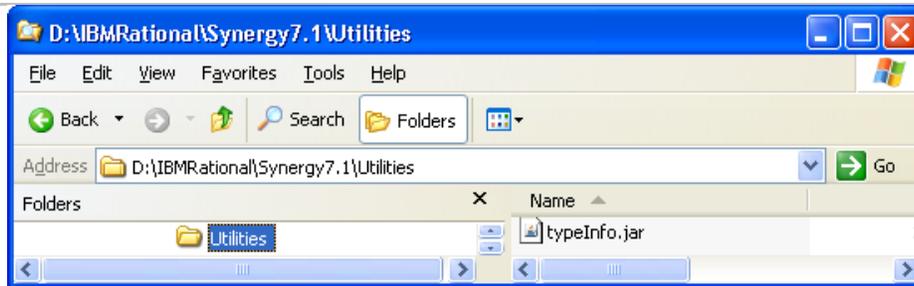


If the names do not match, the menu will not appear.

### 2.8.2 Menu appears but application does not start

Check the IBM Rational Synergy Message View. If there is no error message relating to this plugin, check that script, application or executable that is to be invoked has the same name (and path) as set out in the appropriate part of the `plugin.xml`'s `action` element:

```
<action>
  <exec>java</exec>
  <arg>-jar</arg>
  <arg>%ccm_home%\Utilities\typeInfo.jar</arg>
  <arg>%ccm_home%</arg>
</action>
```



If the names do not match, the menu appears in full, but does not action the script or executable (and does not report an error).

Alternatively, the script or application may have the correct name but does not have the correct functionality or has an error. Look at the source for the script, application or executable and make sure that the error reports (Standard Error, `stderr`) are directed to the console.

### 2.8.3 Menu appears: Error message in Message View or log on clicking menu item

The error report depends on whether you have set up your script, application or executable to direct all error reports (Standard Error, `stderr`) to the console.

Here is a sample error report as shown in the `ccm_client.log` file:

```
02/24 11:53:23 java.lang.NullPointerException
02/24 11:53:23 at
com.telelogic.cm.ccm_client.util.CcmPluginManager.loadPlugin(CcmPluginManager.java:190
)
02/24 11:53:23 at
com.telelogic.cm.ccm_client.util.CcmPluginManager.initializePlugins(CcmPluginManager.j
ava:126)
02/24 11:53:23 at
com.telelogic.cm.ccm_client.CcmClient.startupClient(CcmClient.java:890)
...
```

The most likely cause of this is that there is an internal error in the script, application or executable.

The easiest way to check this (if possible) is to run the script, application or executable directly in a command prompt session or terminal session. If the script, application or executable is running IBM Rational Synergy command line commands, you will have to start a suitable session (in WEB MODE) by hand before trying to run the script, application or executable. You have to provide the command line arguments as set out either in the `exec` element or in the `arg` elements of the `plugin.xml`.

If you have started a Synergy command line session (and set the environment variable `CCM_ADDR`), remember to stop the Synergy session when you have finished your investigation.

## 3 Sample Plugin

To get you started with the IBM Rational Synergy Architecture, and to give you confidence about installing and using the new features, the `TypeAssist.zip` file available with this document provides the `plugin.xml`, `TypeAssist.properties` file and the associated `typeInfo.jar` Java application for a simple (and useful) Synergy plugin. (This plugin provides a single piece of functionality to do with Type handling, but it could be extended.)

This plugin can be added to any IBM Rational Synergy client.

### 3.1 UNIX/Linux

Identify the IBM Rational Synergy installation directory: in the rest of this section, this is shown as `$CCM_HOME`.

Create directory (hierarchy):

```
$CCM_HOME%/etc/plugins/Types
```

Into this directory copy the files:

```
plugin.xml
TypeAssist.properties
typeInfo.jar
```

You may need to change the `plugin.xml` and `TypeAssist.properties` to make them appear correct in certain UNIX editors (`vi`, for example):

```
cd $CCM_HOME%/etc/plugins/Types
dos2unix plugin.xml
dos2unix TypeAssist.properties
```

You need to edit the `plugin.xml` file before it is usable. In particular, you need to change the Windows folder markers ("`\`") to UNIX directory markers ("`/`");

In the `plugin.xml` file, change:

```
<exec>%ccm_home%\jre\bin\java</exec>
to
<exec>%ccm_home%/jre/bin/java</exec>
```

and change

```
<arg>%plugin_path%\typeInfo.jar</arg>
to
<arg>%plugin_path%/typeInfo.jar</arg>
```

Check the permissions on each of the files.

Start up the IBM Rational Synergy client. You now see the new menu "Types" in the application main menu bar.

## 3.2 Windows

Close down any IBM Rational Synergy client which is running.

Identify the IBM Rational Synergy installation directory: in the rest of this section, this is shown as %CCM\_HOME%.

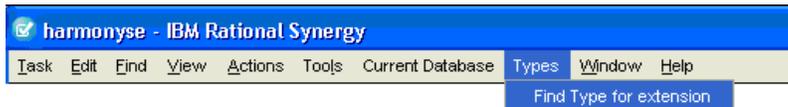
Create folder (hierarchy):

```
%CCM_HOME%\etc\plugins\Types
```

Into this folder copy the files:

```
plugin.xml  
TypeAssist.properties  
typeInfo.jar
```

Start up the IBM Rational Synergy client. You now see the new menu “Types” in the application main menu bar.



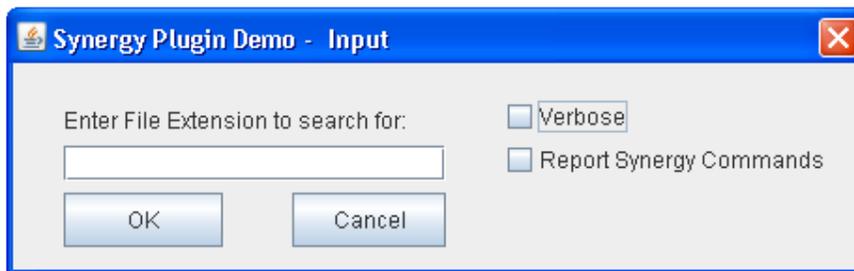
## APPENDIX: Type Finder plugin

This Appendix is a “User Guide” for the IBM Rational Synergy “Type finder” plugin which is used in the examples in this Guide.

When you are going to migrate source code into IBM Rational Synergy – either by means of the Migrate functionality or by accepting work area changes when the work area has new source materials copied into it – you need to know that there are suitable type definitions set up in IBM Rational Synergy for all the new types of source that you need to migrate.

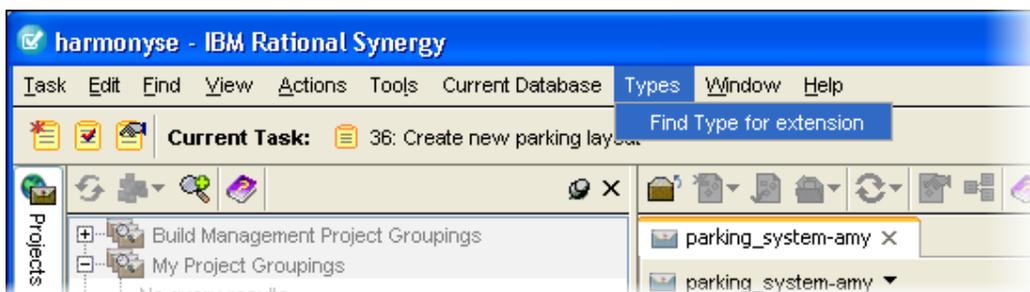
You can check on the types using a Migrate Preview, of course, but this plugin shows you if your selected file type is recognized by IBM Rational Synergy and if so, what type does it map onto.

*Note* that this plugin has been developed for demonstration purposes and although it is a fully working plugin, it is not fully featured.



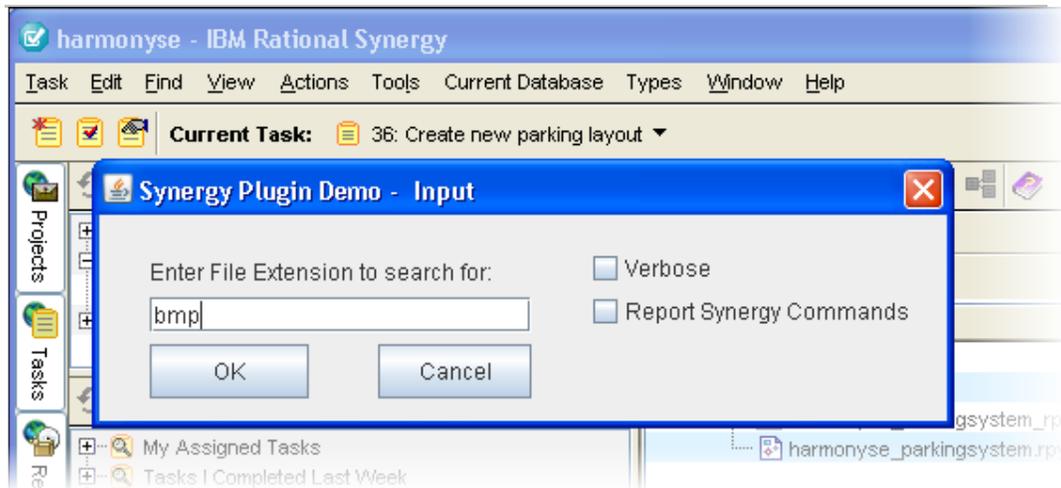
When you have set up the `plugin.xml` and the `TypeAssist.properties` files in the `$CCM_HOME/etc/plugins/Types` directory (`%CCM_HOME\etc\plugins\Types` folder in Windows), restart the IBM Rational Synergy client.

You now get a new menu in the IBM Synergy Menu bar: **Types**. The menu has a single submenu item: **Find Type for Extension**.



(In this screenshot, there is another plugin already installed – the Database Documenter for the current database).

Click the menu option **Find Type for Extension**. The **Input** (prompt) dialog box is displayed.



Enter the extension to search for (do *not* include the period ( , ) preceding the extension).

Click **OK**.

The application compares the extension to the strings in the Windows file match patterns for each appropriate Synergy type definition. If a match is found, the type is reported in the Report dialog. If no match is found, this is reported.

## Runtime Options

### Verbose

This provides verbose messages to help identify what is happening in those cases where the plug in application is not working appropriately.

The setting of this option is *not* persistent: you must select it explicitly on each use of the plugin (should you need to use it).

### Report Synergy Commands

If you check this option, each Rational Synergy command that is run is shown in the Rational Synergy Message View.

The Synergy commands can be copied from the Message View if needed, to check on the commands and their return values.

The setting of this option is *not* persistent: you must select it explicitly on each use of the plugin (should you need to use it).

## Buttons

**Cancel**      Quits the IBM Rational Synergy Type Finder plugin utility.

**OK** The application uses the current IBM Rational Synergy session (current user on current database) to look at the WINDOWS migration match strings (the `win_file_match` attribute) for each type definition..

*Note* that this demo is not full featured – it does not, for example, analyze the equivalent UNIX string matches.

The application then compares the extension entered by the user to the strings in the Windows file match patterns. If a match is found, the type is reported to you in the Report dialog.

During the operation of the plug in, status information is reported in the IBM Rational Synergy Message View (and in the `ccm_client.log` file).



If you have opted to see Verbose messaging, the pattern matching sequences (and other useful debug messages) are shown in the Message View.

If you have opted to Report Synergy Commands, each Synergy command is shown in the Message View as it is run.

The following screen capture shows the IBM Rational Synergy Message View when both the above options are selected when identifying the Synergy type for a specific extension. *Note* that all information sent to the Synergy Message View is also written to the `ccm_client.log` file.

```

Messages - harmonyse - IBM Rational Synergy
User demo on database \\tls2009\ccmdb\harmonySE using server http://tls2009:8400
.\. [Eb] [Mm] [Pp] $

Executing Synergy Command: D:\IBMRational\Synergy7.1\bin\ccm delimiter

Executing Synergy Command: D:\IBMRational\Synergy7.1\bin\ccm query -u "cvtype = 'cvtype'" -f %objectname

Executing Synergy Command: D:\IBMRational\Synergy7.1\bin\ccm attr -s win_file_match cvtype-l:cvtype:AC

Executing Synergy Command: D:\IBMRational\Synergy7.1\bin\ccm attr -s win_file_match ascii-l:cvtype:base
.\. [Eb] [Mm] [Pp] $::.\. [Dd] [Ee] [Ff] $
.\. [Eb] [Mm] [Pp] $::.\. [Ff] $
.\. [Eb] [Mm] [Pp] $::.\. [Ff] [Oo] [Rr] $
.\. [Eb] [Mm] [Pp] $::.\. [Ii] [Nn] [Ii] $
.\. [Eb] [Mm] [Pp] $::.\. [Rr] [Cc] $
.\. [Eb] [Mm] [Pp] $::.\. [Tt] [Xx] [Tt] $
.\. [Eb] [Mm] [Pp] $::

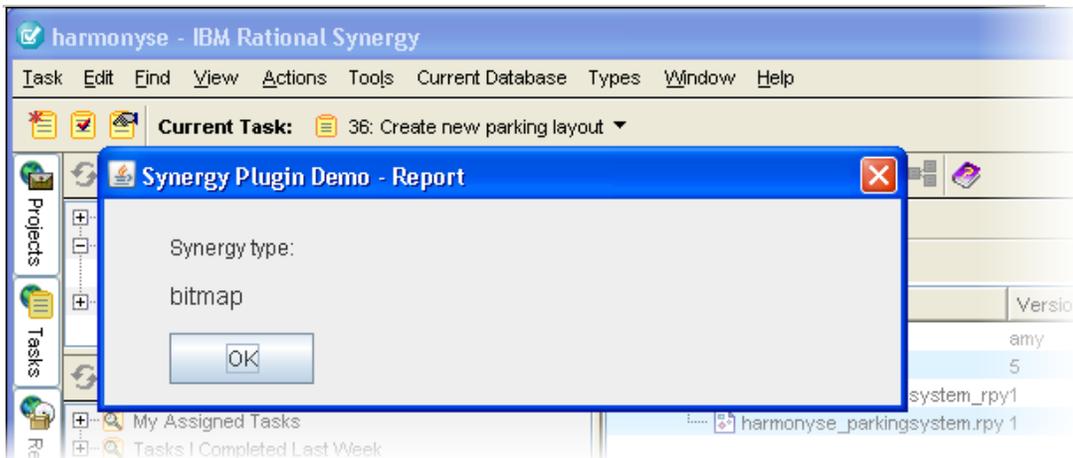
Executing Synergy Command: D:\IBMRational\Synergy7.1\bin\ccm attr -s win_file_match binary-l:cvtype:base
.\. [Eb] [Mm] [Pp] $::.\. [Bb] [Ss] [Cc] $
.\. [Eb] [Mm] [Pp] $::.\. [Cc] [Uu] [Rr] $
.\. [Eb] [Mm] [Pp] $::.\. [Dd] [Oo] [Cc] $
.\. [Eb] [Mm] [Pp] $::.\. [Hh] [Ll] [Pp] $
.\. [Eb] [Mm] [Pp] $::.\. [Ii] [Cc] [Oo] $
.\. [Eb] [Mm] [Pp] $::.\. [Ii] [Ll] [Kk] $
.\. [Eb] [Mm] [Pp] $::.\. [Pp] [Cc] [Hh] $
.\. [Eb] [Mm] [Pp] $::.\. [Pp] [Dd] [Bb] $
.\. [Eb] [Mm] [Pp] $::.\. [Ss] [Bb] [Rr] $
.\. [Eb] [Mm] [Pp] $::.\. [Vv] [Cc] [Pp] $
.\. [Eb] [Mm] [Pp] $::.\. [Ww] [Rr] [Ii] $
.\. [Eb] [Mm] [Pp] $::.\. [Ww] [Aa] [Vv] $
.\. [Eb] [Mm] [Pp] $::

Executing Synergy Command: D:\IBMRational\Synergy7.1\bin\ccm attr -s win_file_match bitmap-l:cvtype:base
.\. [Eb] [Mm] [Pp] $::.\. [Bb] [Mm] [Pp] $
Synergy type is: bitmap

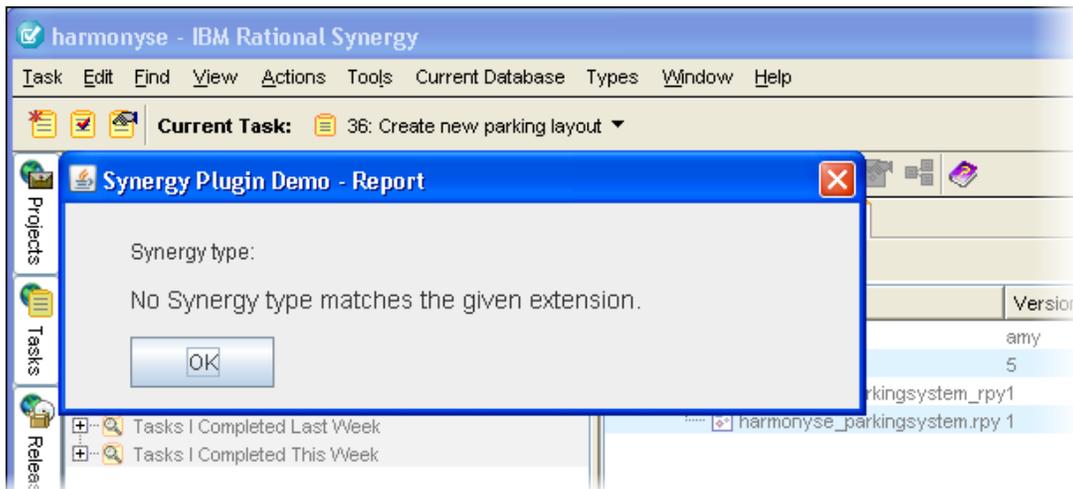
```

*Note* that during the operation of the plugin itself, there is no explicit GUI display (no progress bar, for example): the Input (prompt) dialog disappears immediately and the Report (result) dialog is displayed when either a match is found or the processing has completed without finding a match. The reason for this is that the plugin is processing Synergy information which is unlikely to affect the ongoing Synergy development and build management work. If you want to see “progress”, use the options provided on the start dialog and display the IBM Rational Synergy Message View.

If a match is found, the IBM Rational Synergy type associated with the match is reported (and processing stops at this point):



If no pattern match is found, this is reported at the end of all the processing:



*Note* that if a specific extension is used for files managed by Synergy within a UNIX environment only (and for UNIX file migration), this is not reported in this demonstration plugin. Under these circumstances, the plugin reports that no Synergy types match the given extension.

## Notes and warnings

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send written license inquiries to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send written inquiries to:  
IBM World Trade Asia Corporation

Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions. Therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational Software  
IBM Corporation  
1 Rogers Street Cambridge,

---

Massachusetts 02142  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information as softcopy, the photographs and color illustrations may not appear.