Eresia File Injection Portal (FIP) Version 2.1

CML00037-21

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com

August 23, 2010

Contents

1 Introduction

1.1 Overview

The File Injection Portal (FIP) is a Graphical User Interface (GUI) together with a number of elements to which Eresia Thistle scripts can interact.

Within the GUI, a Workspace allows the user to manipulate Records, view logs containing these Records and record scripts for the Eresia Test Environment.

Customers can configure or have configured meta-data objects using the Code Magus object types[1] schema which uses, for example, COBOL copybook layouts. When used in the GUI, these meta-data objects can be used to view, edit, search and filter file records in terms of the elements of the meta-data. Additionally, the GUI supports querying the contents of files in terms of arbitrary expressions over the records expressed in terms of the meta-data elements.

On exit from the GUI, the FIP saves all information in the Workspace to a user named file, which can be re-opened later in order to continue working.

In order to connect to files and data sources (e.g. query result sets), a file abstraction layer is used. Components that implement the interface to this layer are called Access Methods. There are a number of standard Access Methods available, and an interface description and samples are included allowing the customer to implement their own access methods.

When used for testing, FIP scripts are executed within Eresia.

The GUI of FIP is divided into two sections. The first section contains a hierarchical view of the elements contained in FIP, and the second section is used to display information to the user.

The first section is divided into two columns. The first column contains the name of the element in the hierarchy and the second column contains the value of the element.

From section 3 on page 9 is the start of the reference of all of the element of the visual part of the FIP. Information is provided on how all of the individual elements of FIP work.

1.2 Record Types in the GUI

1.2.1 Apparent types

All records that appear in the GUI are processed over an object types definition file, and the resulting object type is displayed in the right hand column next to the record. This processing is done automatically, and it is always done for any buffer in the GUI.

This type is the apparent type of the record.

1.2.2 Set types

A record can also be set to a specific type, and when this is done, the record will always behave as though it is an instance of the set type, even if the record has an apparent type which is different from the set type.

Any usage of the record will reflect that the record is of the set type. As mentioned, the set type supersedes the apparent type of a record.

See sub-section 4.11 on page 17 for details on how to assign the set type to a record.

1.3 Files, Records and the FIP

The purpose of the FIP is to manipulate and make use of logged records that conform to Object Type Definition Files and the corresponding Cobol copybooks. These Object Type Definition Files (called Object Type Definition Files in the FIP) define which fields are used to determine the type of a Record using the fields and layouts of records defined in the copybooks.

The FIP allows the user to view which Records are described in the Object Type Definition Files. These Records can be inspected so that the user can view the entire hierarchical view of a Record including all of the fields contained within that Record Type.

From here the user can create a new Record using one of these Record Type.

The FIP also allows the user to view an entire buffer of Records stored in a file or a log (known as a Candidate Data File). The FIP will type these Records automatically so that every Record within the buffer can be inspected and the real data contained within the Record can be viewed on field-by-field basis, or as an entire block of data. This file or log is read with a Open Spec which is a program designed to read the Records in a log file of a particular format, and to load these Records into the FIP. See section 2.6.2 on page 9 for further details.

Once the user has real Records, the data within each Record can be updated or changed, and saved to a Workspace, which can later be loaded.

1.4 summary

In summary, FIP can be used to:

- View Records in a tree form
- Edit and change the content (i.e. the data) in a Record

- Create a new Record from an existing Record Type
- Use an Open Spec to read a Candidate Data File containing a log of these Records

2 Elements of the FIP

2.1 Overview

The FIP uses a number of different element which the user interacts with. These elements all have very specific roles in the GUI. Elements are represented in a tree structure, which allows the user to easily determine the relationships of each of these elements to each other.

For example, the 'Record Collection' element, which contains a collection of Record Types as the name implies, will contain each of its Record definitions under the Record Collection element, and the 'Worksheet Collection' element will contain a collection of all of the Worksheets in the Workspace.

This chapter will explain what purpose these elements serve, and how these elements are related. The technical usage of the elements is given starting with section 3 on page 9.

2.2 The Workspace

The Workspace can be described as a set of properties, Records and settings which are grouped together in one project.

Each Workspace is described in a single file, and this file has its own grammar to describe the contents of the Workspace. All changes, additions etc, are contained in this Workspace file.

This allows the user to make changes to their Workspace, save it and be able to re-open it at a later stage to continue working with it.

The Workspace contains the following elements:

- Description

  The description is simple text written by the user to describe the Workspace. This is a note to the owner or users of the Workspace, and serves no other purpose.

- The Environment Variables collection.

  The environment variables collection contains a list of environment variables that are set when a workspace is loaded. They are also set when you add a new variable or change the name or the value of an existing variable.

- Object Type Definition File

  The Object Type Definition File specifies the definitions, restrictions and typing information of Records which are available to the Workspace.

  Without this Object Type Definition File, the FIP can not know how to determine the type of a Record within a Candidate Data File or a Worksheet.

  This file is also used to list the available Record Types which can be used to create new Records.

- Record Types Collection

  The Record Types Collection element contains a list of all of the Record Types contained in the Object Type Definition File. These Record Types are displayed in a tree view which allows the user to easily determine which fields a particular Record contains. The Record Types in this section can be used to create a new Record within a Worksheet (See section 2.5 on page 8).

- Query Collection

  The query collection element contains a list of all of the queries stored in the Workspace. Each query is a element itself, and has its own behaviour depending on what type of query it is. (See section 2.3 on page 6).

- Worksheet Collection

  The Worksheet Collection element contains a list of all of the Worksheets used in the Workspace. Each of these Worksheets contains a list of Records used in that Worksheet. See section 2.4 on page 8.

  Each Worksheet contains a Record Collection which in turn contains a list of Records used in that Worksheet. Although Records appear under a Worksheet, they are described in their own section. See section 2.5 on page 8.

- Candidate Data Files

  The Candidate Data Files are actual files containing multiple Records. The Records are embedded within a Candidate Data File in a physical data layout that has its own data format.

  This is why the Candidate Data File element requires an Open Spec which describes the method used to read the data file in the Candidate Data File element. See section 2.6 on page 8.

2.3 Queries

A query element has a name, which is simply a name the user assigns to a query. This name does not have any other purpose in the FIP.

The element can be expanded, and the sub-elements of the query will be different depending on what type of query it is. Currently there are three different types of queries. These are:

1. Record Type Query (include or exclude the given record types)
2. ASCII/EBCDIC Query (include or exclude records that match a given regular expression)
3. Query (include or exclude records that match the specified query).

When a query is created, certain operations can be performed on it. Again, these operations will vary according to what type of query has been created.

2.3.1 Record Type Query

The Record Type query allows the user to list all of the Records inside the Candidate Data File that are in the list of specified queries. The user can choose either an include filter or an exclude filter. Therefore the user can specify that they wish to include all the Records in the Candidate Data File that are listed in the list of Record Types, or they can choose to view all Records in the Candidate Data File, except the ones listed in the list of Record Types specified for the query.

2.3.2 Like ASCII/Like EBCDIC Query

This query will list all Records which match a regular expression. As the name implies, this regular expression can be either ASCII or EBCDIC.

Again the user can choose whether they wish to include or exclude all Records that conform to the given regular expression.

2.3.3 QueryString Query

This query allows the user to use a real Query using the objtypes query language. The user first selects the base Record Type to perform the query on using a right-click on the Record Type element, and then types a query in the WHERE clause.

The user can make use of substitutions in place of field names by specifying wild cards, which are preceded by a dollar sign. When the user uses a substitution, more nodes will be created below the query element. The user can then specify a field name by right clicking on the substitution name and selecting the field name from the list that appears.

2.4 Worksheets

Worksheets are used to allow users to store a list of Records of interest to the user, and as an area where the values of fields can be altered.

Other functions of the worksheet are:

- Storing a list of Records
- Creating new Records using the Record Types Collection
- Creating copies of Records using the Candidate Data File
- Recording a script from the Eresia Test Environment

Each Workspace has a worksheet collection under which the Worksheets are stored.

The user can add a Worksheet to the Worksheet Collection, and when the Workspace is saved, the list of Records inside the Worksheet are also saved to the Workspace file.

2.5 Records

2.5.1 Viewing Record entries

A Record entry is displayed as a tree structure, showing each field within the Record. The Record can be expanded to the leaf level to view all data within the buffer.

2.5.2 Editing and changing Record buffers

The user is able to edit records simply by double-clicking the value of a field and entering the new value. The buffer of the Record will be updated when this is done. .

Once a worksheet has been created, the user can add Records to the Worksheet. This Record can be created from the Record Types Collection, or from a Candidate Data File, or copied from another Record in any Worksheet.

When a Record is created from the Record Types Collection, a Record is generated from and conforms to the Record defined by the Object Type used to create it.

Examples of the usage of Records is explained in detail in the technical sections starting with section 3 on page 9.

2.6 Candidate Data Files

Candidate Data Files hold buffers of Records within a physical data structure which is accessed via the Open Spec object.

2.6.1 Data file

As mentioned earlier, the Candidate Data File itself is a disk file which contains a buffer of Records. The Candidate Data File has its own data format, and therefore needs a program to decode this information. This is the role of the Open Spec (see section 2.6.2 on page 9).

The Candidate Data File value refers to the physical path of the actual file.

2.6.2 Open Spec object

The Open Spec object provides information to the Open Spec library on how to read and interpret a Candidate Data File and extract the Records from the physical structure.

3 The Graphical User Interface

The Graphical User Interface (GUI) is a visual representation of the contents of a Workspace. Each of the elements that are visible in the GUI will be described in their own section (See section 4 on page 11).

The GUI is split into two sections. These sections are divided by a line down the screen. The section on the left contains the different elements available in the Workspace, which are the Description, Record Types Collection, Query Collection, Worksheet Collection and Candidate Data Files.

These items are all displayed in a tree structure. In this way it is easy to tell how the elements are related to one another.

The section on the right hand side of the line displays the value of the corresponding element on the left. This value can be double-clicked and edited if the corresponding element has a value that can be changed. An example of this is, the 'Description' element of a Workspace, which contains a simple textual description of the project.

There are actions that can be performed on many of these elements. These actions can be performed by right-clicking on the element. A drop-down menu will appear, and will display a list of all of the available actions that can be performed on that particular element.

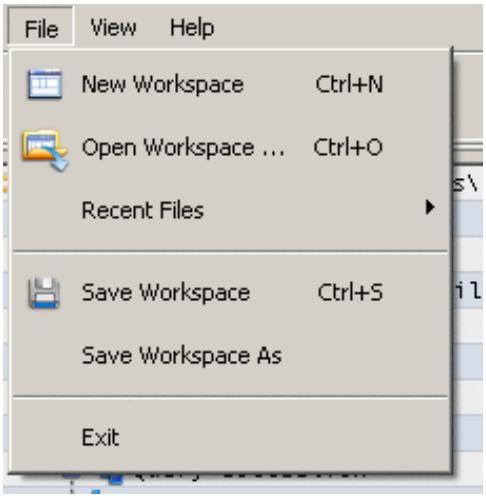
For example, if there is no Description defined for the Workspace, a description can be added by right-clicking on the Workspace element and selecting 'Add' and then 'Description'. A new element will be added under the Workspace element for the description.

These actions will be discussed in detail in their own section.

3.1 The Menu Bar

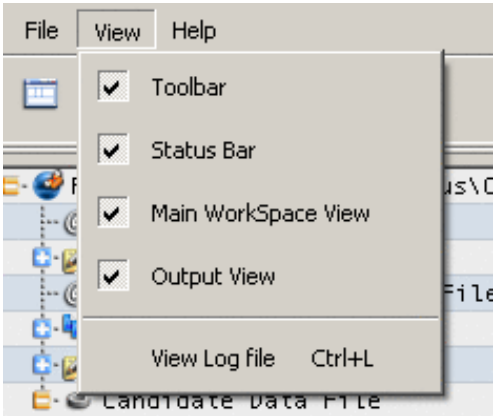
This section explains the layout of the menu bar. Individual menu items are also explained.

3.1.1 File



Actions	
New Workspace	Create a new, empty Workspace
Open Workspace	Open an existing Workspace from file
Recent Files	Display and load a recently opened Workspace
Save Workspace	Save the current Workspace to file
Save Workspace As	Save the current Workspace to a new file
Exit	Close the GUI

3.1.2 View



Actions	
Toolbar	Toggle displaying of the toolbar
Status Bar	Toggle displaying of the status bar
Main Workspace View	Toggle displaying of the Workspace tree
Output View	Toggle displaying of the output window
View Log file	Display the log file of the current session

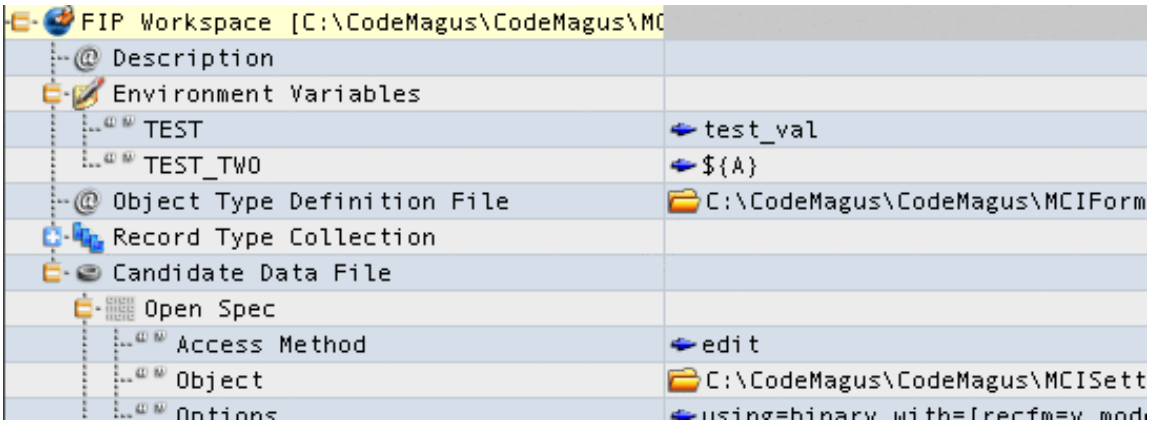
3.1.3 Help

Actions	
Contents and Index	Display the help file for the FIP
About Eresia FIP	Show the Help/About dialogue

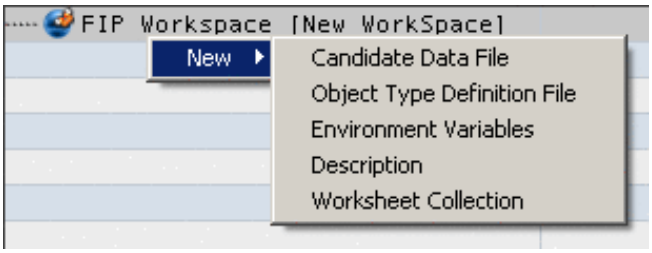
4 GUI Elements

Each element of the GUI is described in this section, along with its actions and some examples of usage. The tables listed below contain descriptions on the actions (right-click options) for each element.

4.1 FIP Workspace

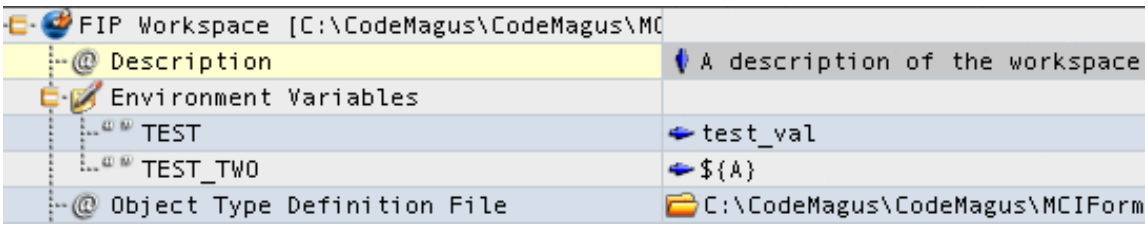


The Workspace element contains all of the other elements pertaining to a project. It, and all the items below it, can be saved to a file; normally with a suffix of ".fpw".



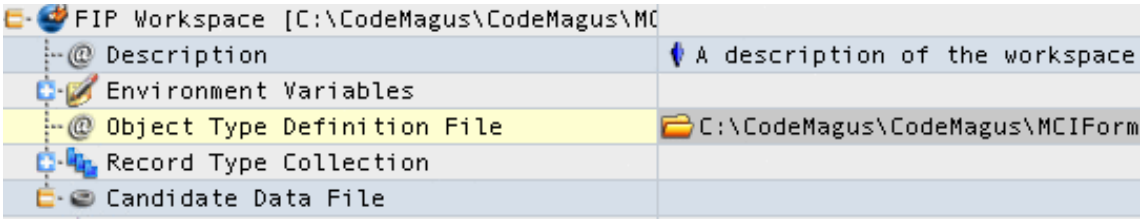
Actions	
New/Description	Add a description of the Workspace
New/Object Types Definition File	Add an Object Type Definition File
New/Query Collection	Add collection of queries to the Workspace
New/Worksheet Collection	Add collection of worksheets to the Workspace
New/Candidate Data File	Add data file containing Records to the Workspace

4.2 Description



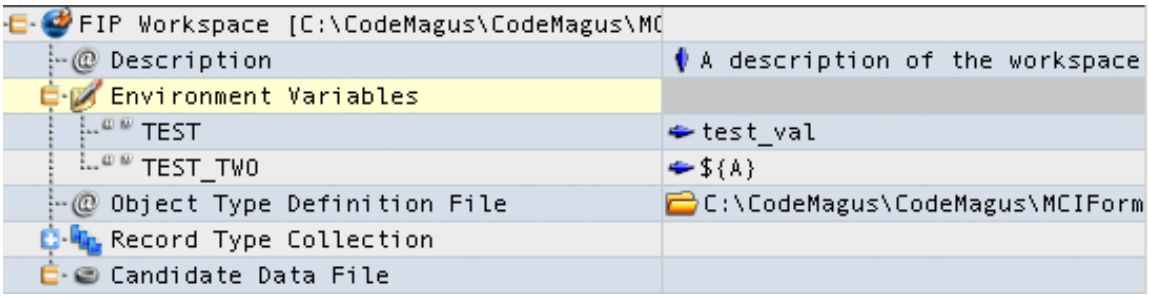
The description simply provides a textual description of the current Workspace.

4.3 Object Type Definition File

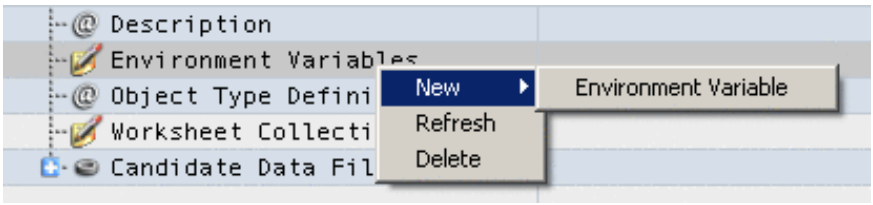


The Object Type Definition File specifies the file which acts as the source of the definition of the Record Types which will be used in the Workspace. When an Object Type Definition File is defined, an entry will appear under the Workspace which is called Record Types Collection. See section 4.6 on page 14.

4.4 Environment Variables collection

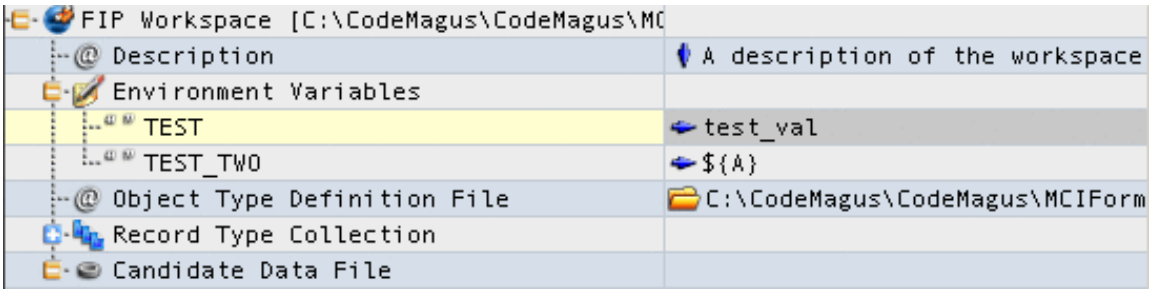


The Environment Variables Collection contains a list of all defined environment variables. Each of these has its own entry. See section 4.5 on page 13.

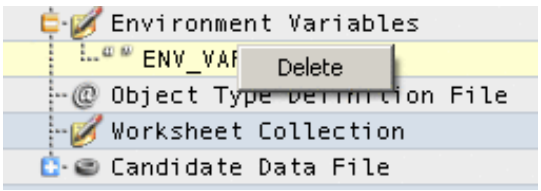


Actions	
New/Environment Variable	Add a new entry
Refresh	Refresh the list of variables
Delete	Remove this element

4.5 Environment Variable

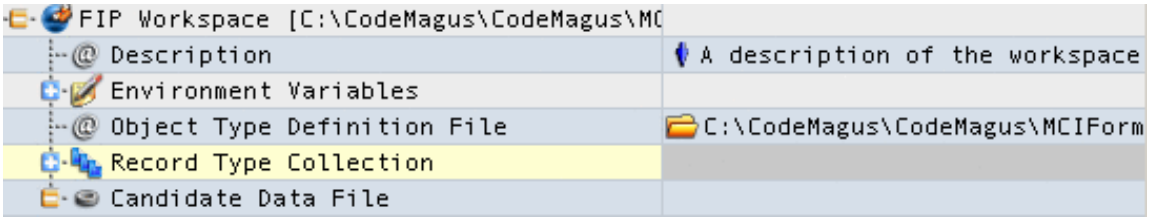


The Environment Variable node has a name and a value. The name is the actual name of the environment variable to set, and the value is the actual value.



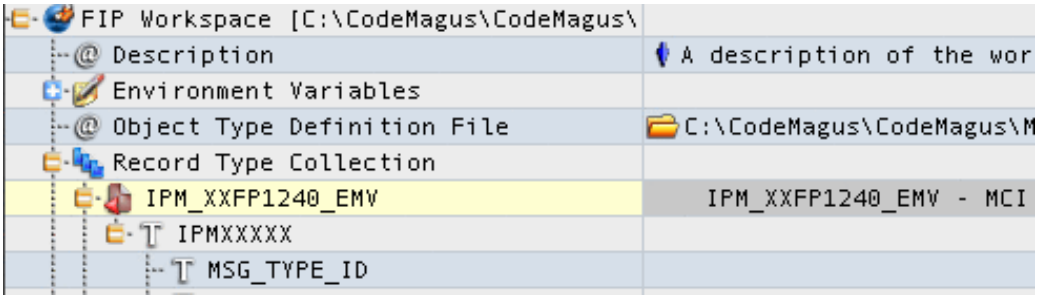
Actions	
Delete	Remove this element

4.6 Record Types Collection

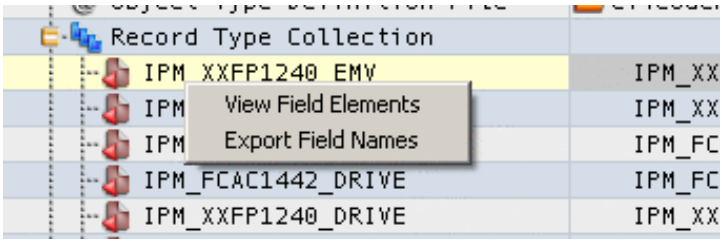


The Record Types Collection contains a list of all defined Record Types within the specified Object Type Definition File. Each of these Record Type entries can be used to get more information about individual types defined in the Object Type Definition File. See section 4.7 on page 14

4.7 Record Type

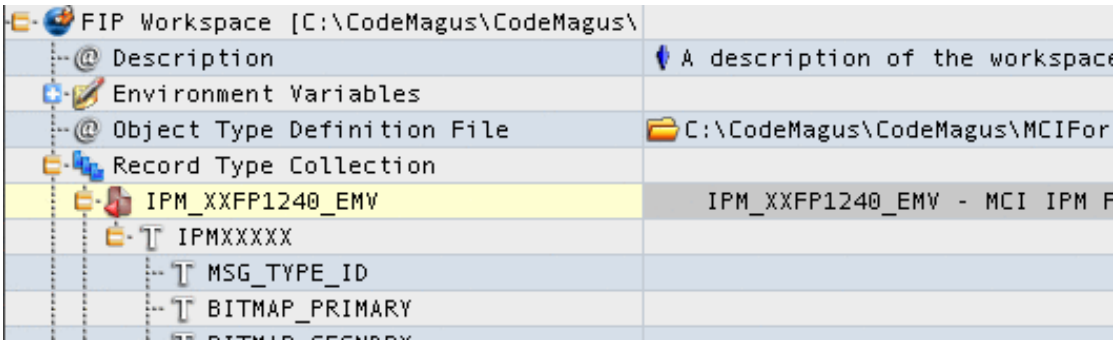


A Record Type element has the type's name, and can be used to view a full Record of that type. These Record Types can be used to create a new, empty Record if a type is dragged to a Worksheet, or if a user performs a Copy/Paste of a Record Type into a Worksheet.



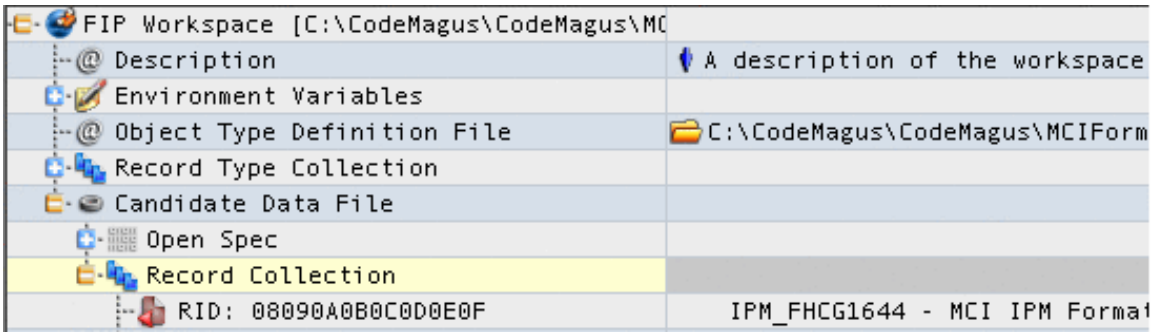
Actions	
View Record Buffer Dump	View the actual buffer of the Record
View Field Elements	Populate the fields of the Record in the tree
Export Field Names	Display a list of field names of the Record

4.8 Record Type Field



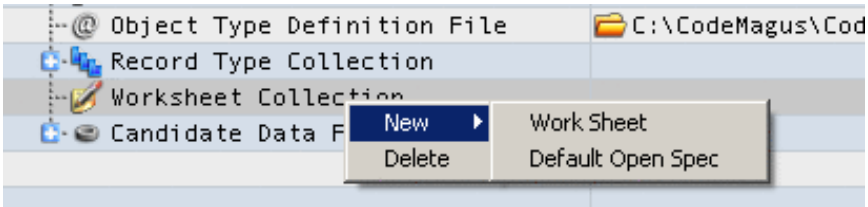
Each field within a Record Type displays the name of the field as it is defined in the Object Type Definition File and copy book. This field shows the name of the field. The value of this entry will always be empty since this element has no buffer. It serves only as a means for the user to view which fields and sub-fields are contained within a given type. See section 4.12 on page 18 for an example of an editable field.

4.9 Worksheet Collection



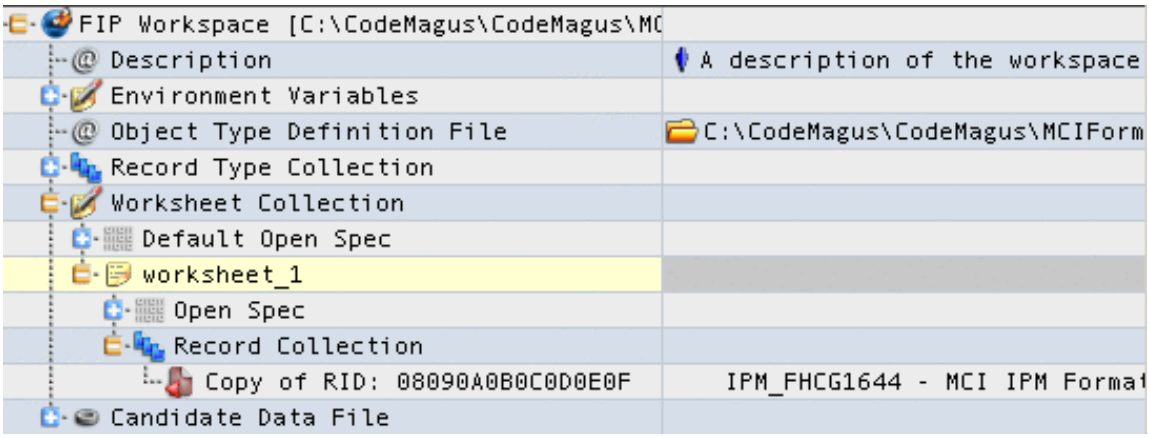
A Worksheet Collection contains a list of Worksheets (see section 4.10 on page 16).

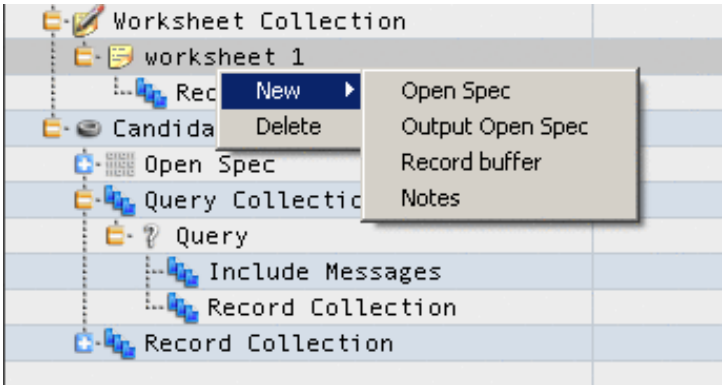
The Worksheet Collection can be used to add new Worksheets to the Workspace, and also to set a Default Open Spec. This Default Open Spec is used when an Open Spec is not set for an individual Worksheet.



Actions	
New/Work Sheet	Add a new Worksheet to this worksheet collection
New/Open Spec	Add a default Open Spec object

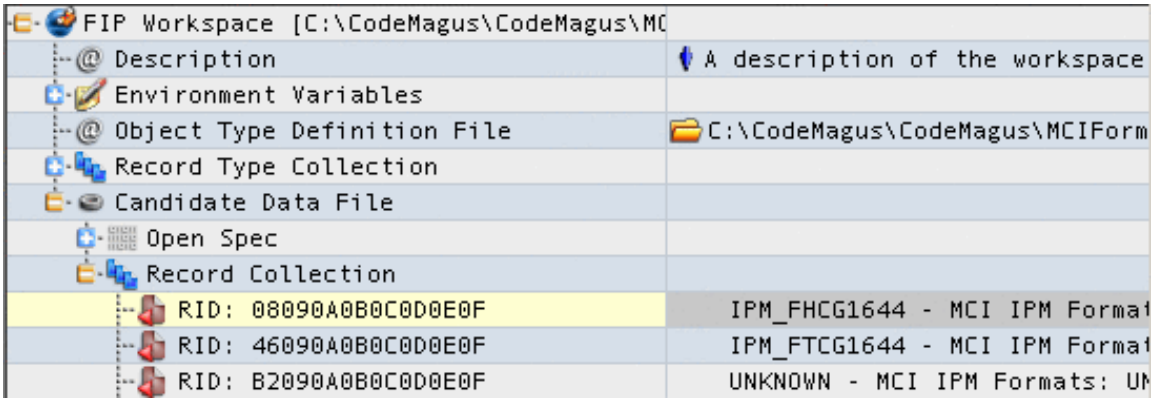
4.10 Worksheet



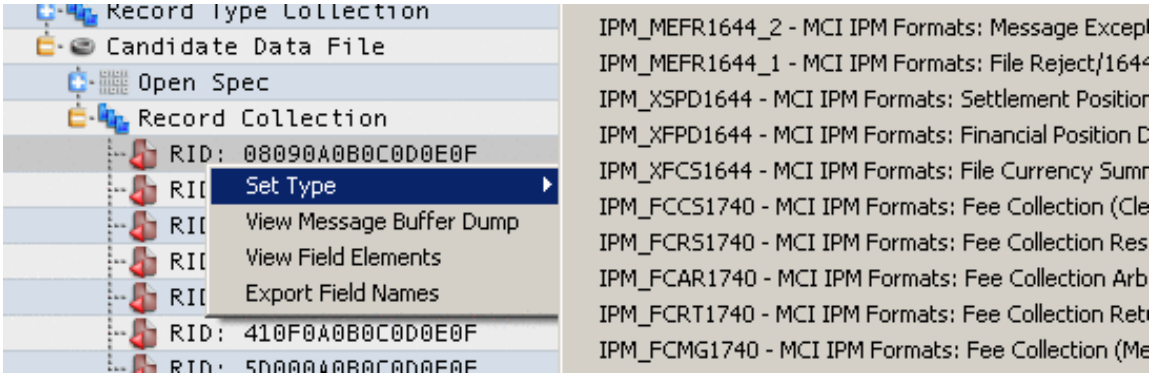


Actions	
New/Record Buffer	Add an empty record buffer to the Worksheet
New/Open Spec	Add an input open spec for recording
New/Output Open Spec	Add an output open spec for writing to file
Delete	Remove this Worksheet from the Worksheet Collection
Record	Record this Worksheet in the Automated Test Environment

4.11 Record



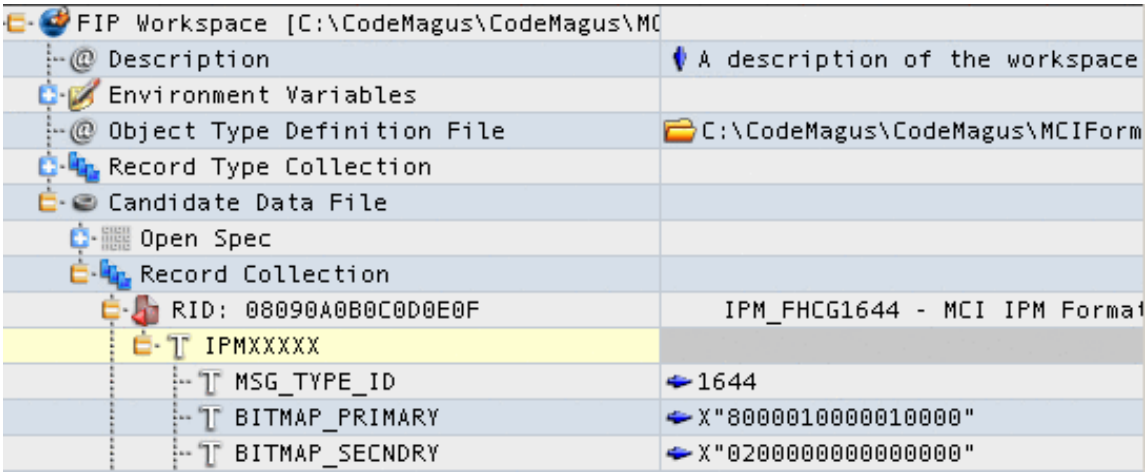
The Record element is a representation of a buffer of data which conforms to a type defined in the object types. This element has different actions available depending on its position in the tree.



Actions	
Set Type	Assign the Set Type of a record to a specific record type
View Record Buffer Dump	View the actual buffer of the Record
View Field Elements	Populate the fields of the Record in the tree
Export Field Names	Display a list of field names of the Record
Delete	Remove this Record from the GUI

Note that only some of these actions will be available to the user depending on where this Record is in the tree and on other conditions.

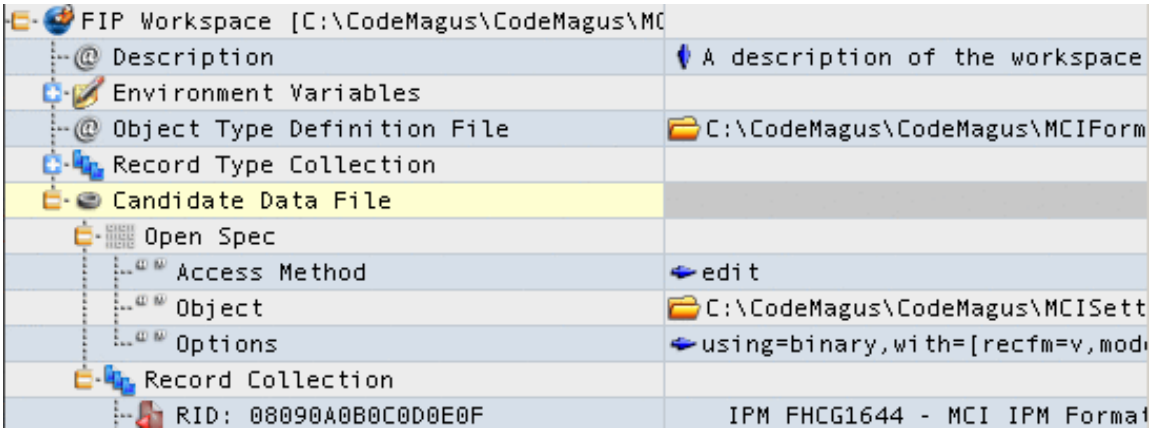
4.12 Record Field



Each field within a Record displays the name of the field as it is defined in the object type and copy book. This field name can not be changed, but the value of the data within this field can be changed if and only if this field is contained within a Record which is in a worksheet. This allows the user to change the data and the actual buffer of the Record.

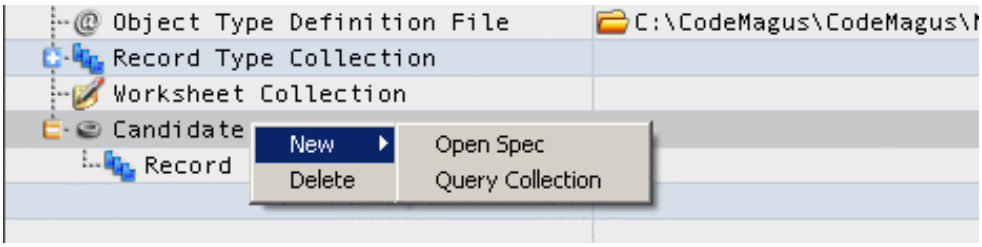
A field may also contain sub-fields. These sub fields can be displayed by expanding the node.

4.13 Candidate Data File



A Candidate Data File is a data file containing a number of Records, in a specific format. This file is read using a Open Spec.

The value of the element on the right hand side is the full path to the actual data file.



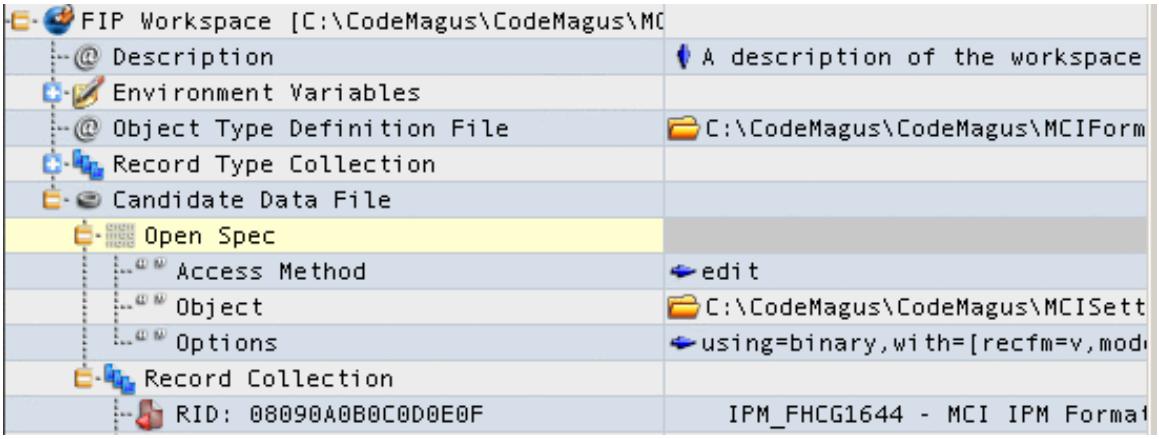
Actions	
New/Open Spec	Add an Open Spec object
New/Query Collection	Add a set of queries

The Open Spec under the Candidate Data File specifies the parameters used to read the data file. If the Open Spec successfully opens and reads the data file, then the messages that are contained inside that data file will be added to the Candidate Data File element under a 'Message Collection' element. See section 4.14 on page 19

Each of these can be viewed, displayed and used to generate new messages in a Worksheet, but can not be edited or changed (See section 4.11 on page 17).

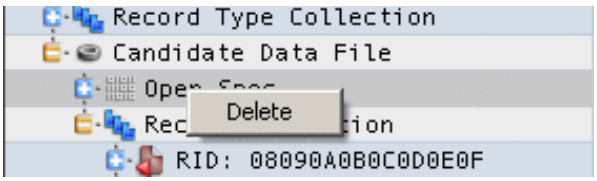
The Candidate Data File can also contain a Query Collection (see section 4.15 on page 21), which contains a list of queries that can be executed on this Candidate Data File.

4.14 Open Spec



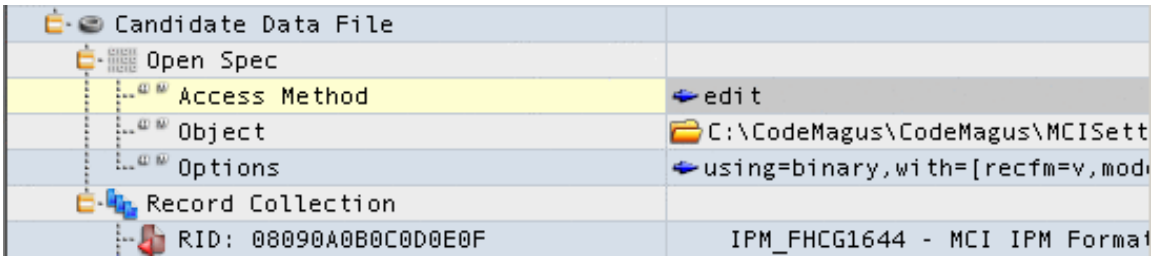
This element simply allows the user to set up the parameters to access the Candidate Data File (see section 4.13 on page 18).

The Open Spec has three properties. The Access Method property defines which access method to use to read the data file. The Object property specifies the full path of the data file. And the Options property is used to define the parameters to pass to the Open Spec library. Further details are supplied in the sub-sections that follow.



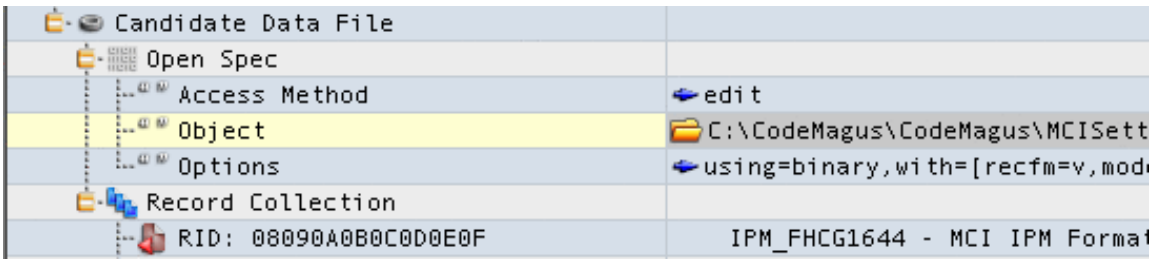
Actions	
Delete	Remove this Open Spec from its parent

4.14.1 Access Method



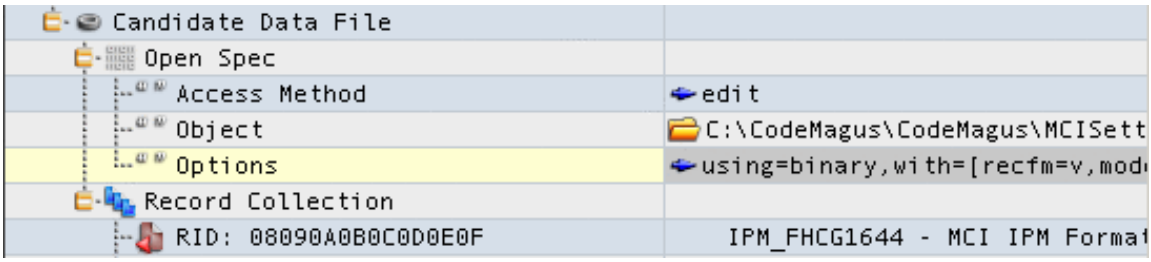
This element is used to specify which access method to use to read from or write to a data file (log). The name supplied here specifies the name of the AMD file to use to look up the required information. Some options for this item are FILE, BINARY and NIPLOG.

4.14.2 Object



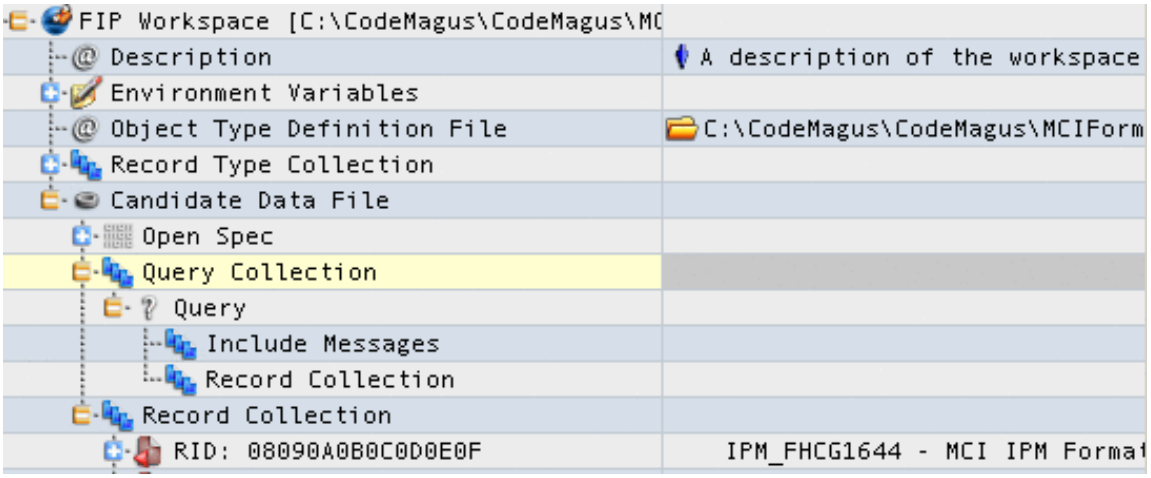
This element is used to specify the full path of the Object which the Open Spec will access using the specified Access Method.

4.14.3 Options



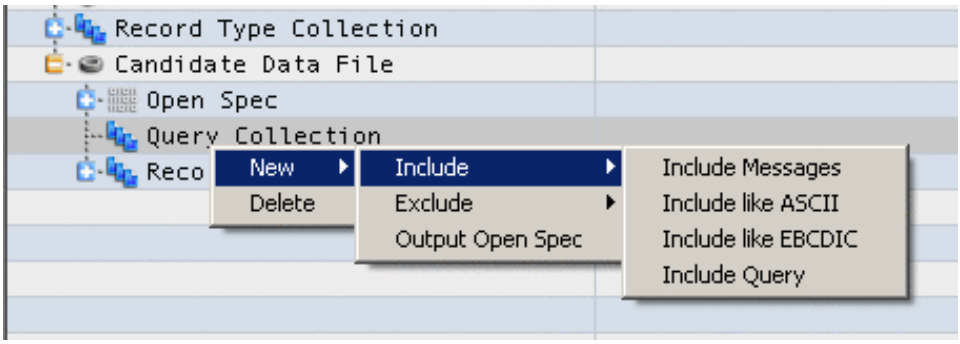
This element is used to specify the full set of options passed to the Open Spec when it is being opened for read or write access.

4.15 Query Collection



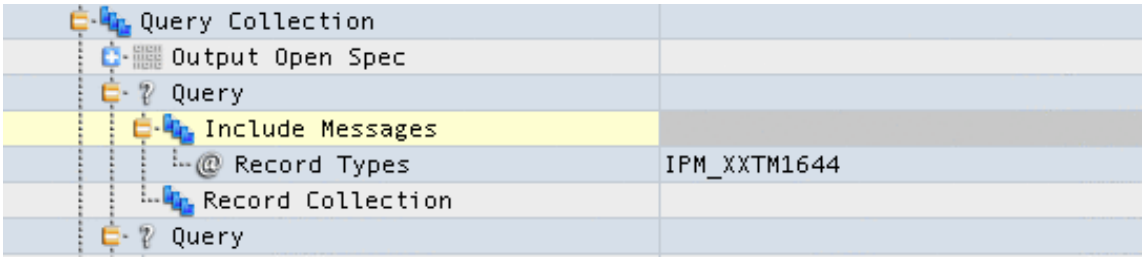
This element simply allows the user to add queries that can be performed on the Candidate Data File. See sections section 4.16 on page 22, section 4.17 on page 23 and section 4.18 on page 23 for more information on these query types.

Note that when a Query Collection node is a child of the Workspace node, then no menu items will be accessible. The use of this collection in the workspace is simply to store queries. Queries can be copied here by copy/paste from other query collections.

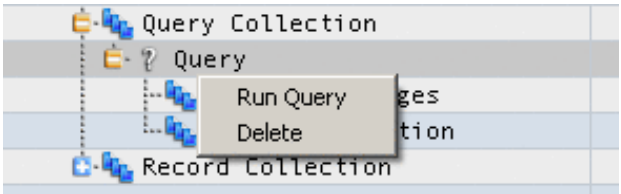


Actions	
New/Include/Records	Add a Record types include filter
New/Include/like ASCII	Add an ASCII include filter
New/Include/like EBCDIC	Add an EBCDIC include filter
New/Include/Query	Add an include Query
New/exclude/Records	Add a Record types exclude filter
New/exclude/like ASCII	Add an ASCII exclude filter
New/exclude/like EBCDIC	Add an EBCDIC exclude filter
New/exclude/Query	Add an exclude Query

4.16 Include Records Query

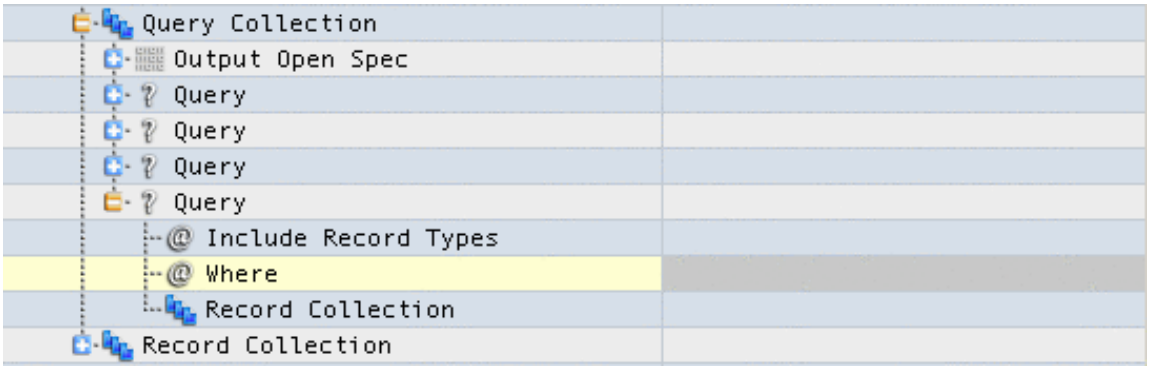


This Query allows the user to filter the Records in a Candidate Data File by the Record Type. The user can either include or exclude a list of Record Types. The user can add to this list by right-clicking on the node 'Record Types' and selecting the Record Type from the list.



Actions	
Run Query	Perform the query and populate the Record list
Save records to file	Write the results of the query to file
Delete	Remove this query from the query collection

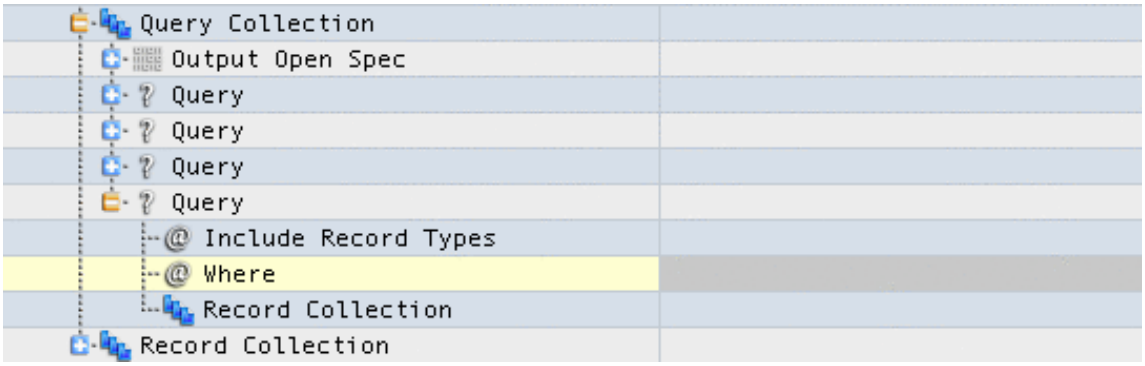
4.17 Include ASCII/EBCDIC Query



This Query allows the user to filter the Records by performing a regular expression on the buffer of the Record. The user specifies this filter by setting the value of the search in the value of the 'Include ASCII' or 'Include EBCDIC' field.

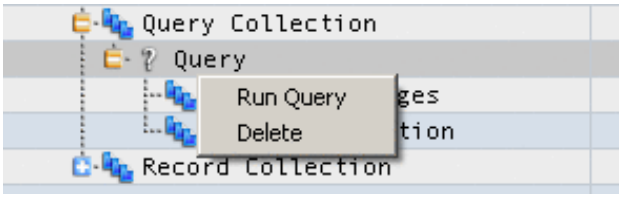
Actions	
Run Query	Perform the query and populate the Record list
Save records to file	Write the results of the query to file
Delete	Remove this query from the query collection

4.18 Include Query



This Query allows the user to filter the Records by performing a query on the Candidate Data File by using the `expeval` library. The user selects the Record Type by right-clicking on the Include Record Type node and selecting the type from the list.

If the user specifies any substitutions in the search string, another node will appear under Substitutions. The user can then right click on each substitution and select the field name from the list.

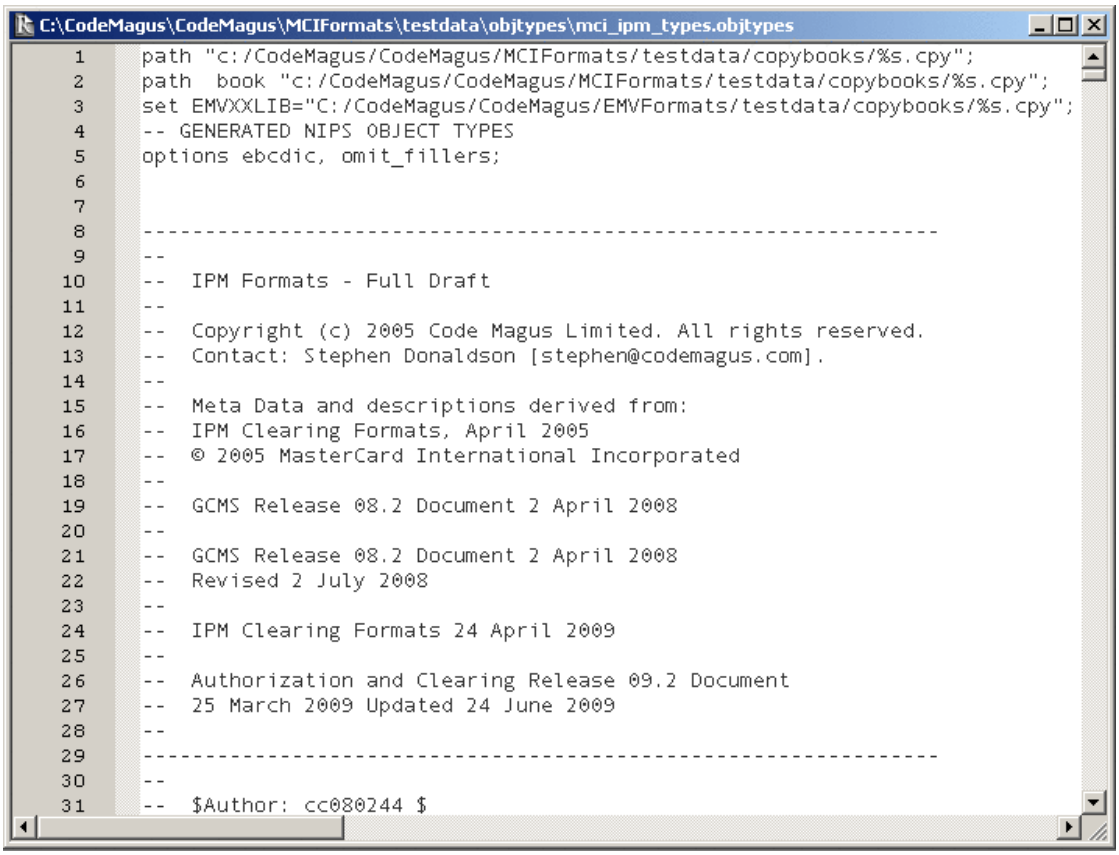


Actions	
Run Query	Perform the query and populate the Record list
Save records to file	Write the results of the query to file
Delete	Remove this query from the query collection

5 GUI Output Windows

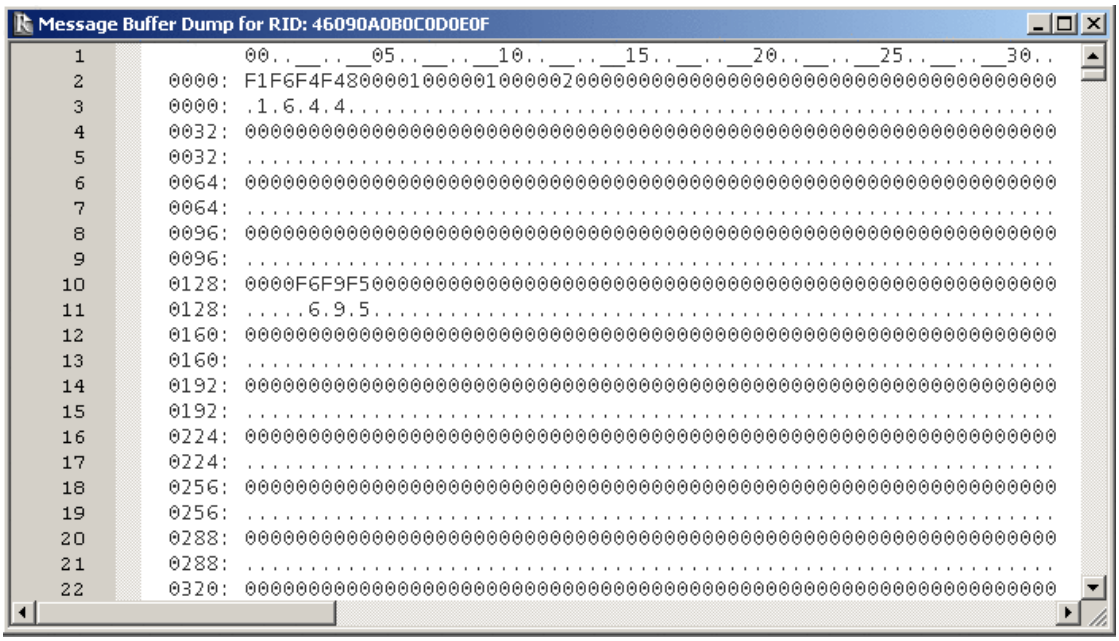
5.1 Object Type Definition File

This window displays the Object Type Definition File.



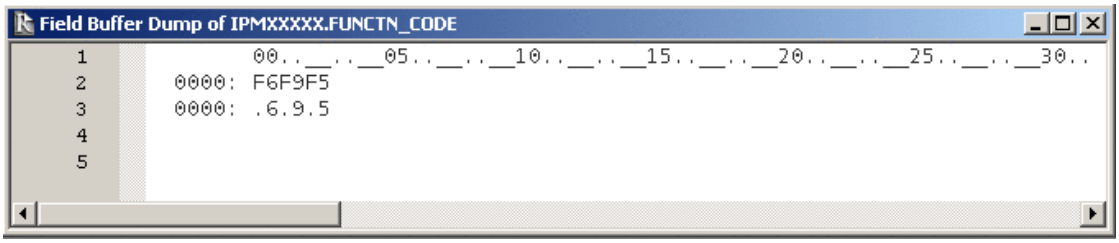
5.2 Message Buffer Dump

This window displays a the buffer of a given message.



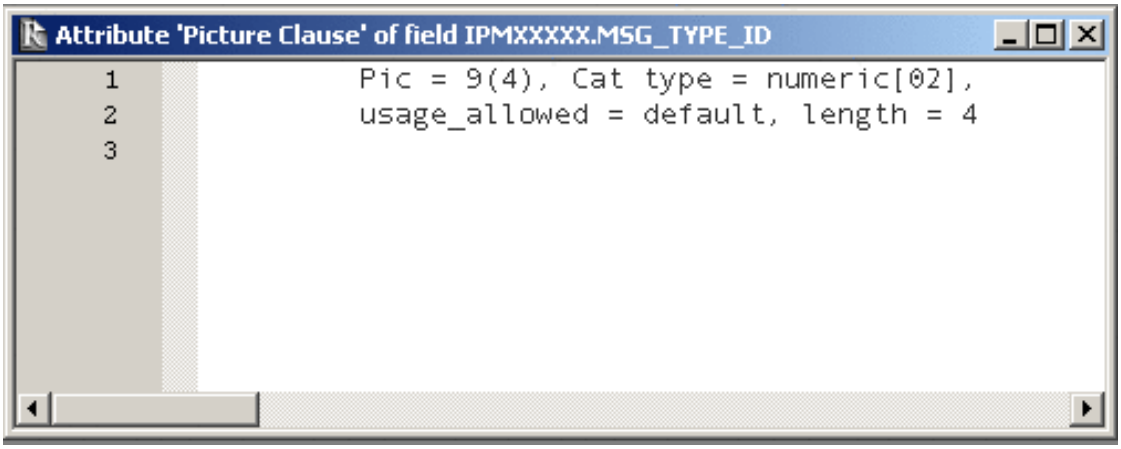
5.3 Field Buffer Dump

This window displays the buffer of the a given field of a message.



5.4 Field Picture Clause

This window displays the picture clause of the given field.

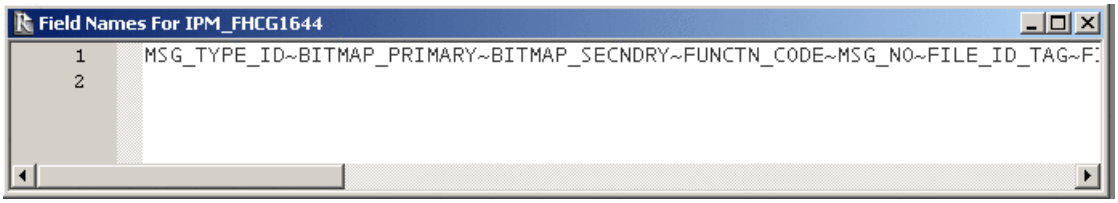


5.5 Field DESCRIPTION clause

This window displays the DESCRIPTION of the given field according to the copybook.

5.6 Exported Field Names

This window displays the names of all the fields within a given message.



References

[1] **objtypes**: Configuring for Object Recognition, Generation and Manipulation. CML Document CML00018-01, Code Magus Limited, July 2008.
http://www.codemagus.com/documents/objtpuref.pdf.