



Eresia Network Injection Portal (NIP) Version 2

CML00038-02

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com
Copyright © 2009 Code Magus Limited
All rights reserved



January 5, 2010

Contents

1	Introduction	3
1.1	Overview	3
1.2	Network Messages and the NIP	3
1.3	Message Types in the GUI	4
1.3.1	Apparent types	4
1.3.2	Set types	4
1.4	summary	4
2	Elements of the NIP	5
2.1	Overview	5
2.2	The Workspace	5
2.3	The Environment Variables collection	5
2.4	The Message Types collection	6
2.5	The Query Collection	6
2.5.1	Message Type Query	6
2.5.2	Like ASCII/EBCDIC Query	6
2.5.3	Object Types Query	7
2.6	The Candidate Log File(s)	7
2.7	The Worksheet Collection	7
2.7.1	The Network Control Program	8
2.7.2	Viewing Message entries	8
2.7.3	Editing and changing Message buffers	8
2.7.4	Sending Messages over a network	8
3	The Graphical User Interface	9
3.1	The Menu Bar	9
3.1.1	File	9
3.1.2	View	10
3.1.3	Help	10
4	GUI Elements	11
4.1	NIP Workspace	11
4.2	Description	11
4.3	Object Types Definition File	11
4.4	Environment Variables collection	12
4.5	Environment Variable	12
4.6	Message Types Collection	13
4.7	Message Type	13
4.8	Message Type Field	13
4.9	Worksheet Collection	14
4.10	Worksheet	14
4.11	Network Control Program	15
4.11.1	NCP Parameter	15
4.11.2	NCP Attribute Group	16
4.11.3	NCP Attribute	16

4.12	Message	17
4.13	Message Field	17
4.14	Candidate Log File	18
4.15	Open Spec	19
	4.15.1 Access Method	19
	4.15.2 Object	19
	4.15.3 Options	20
4.16	Query Collection	20
4.17	Queries	21
	4.17.1 Include/Exclude Messages Query	21
	4.17.2 Include/Exclude ASCII/EBCDIC Query	22
	4.17.3 Include/Exclude Query	22
5	GUI Output Windows	23
5.1	Object Type Definition File	23
5.2	Message Buffer Dump	23
5.3	Field Buffer Dump	23
5.4	Field Picture Clause	24
5.5	Field DESCRIPTION clause	24
5.6	Exported Field Names	24

1 Introduction

1.1 Overview

The Network Injection Portal (NIP) 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 network messages, view logs containing these messages and record scripts for the Eresia Test environment.

Messages can be viewed, edited, created, deleted and sent over a network to a server, from which the response may also be viewed.

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

The GUI of the NIP is divided into two sections. The first section contains a hierarchical view of the elements contained in the NIP 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.

Section 3 on page 9 onwards contains a reference of all of the elements of the NIP. Information is provided on how all of the individual elements of the NIP work.

1.2 Network Messages and the NIP

The purpose of the NIP is to manipulate and make use of network messages. These messages are identified and mapped by Object Type Definitions defined in Object Type Definition Files and the corresponding COBOL copybooks. These Object Type Definitions define which fields are used to determine the type of a message. The entire message is mapped using the fields and layouts defined in the copybooks and can be viewed as a hierarchy of messages and fields. New messages can be created from one of the defined message types. The Object Type Definitions can be viewed in the NIP.

An external file (or log) containing messages and identified through a Candidate Log File may be viewed in the hierarchy of the NIP Workspace. These messages are automatically typed using the Object Types Definition File named for the Workspace. This message log is loaded into the NIP via an Open Spec library.

In the NIP, messages may also be sent to, and the response received from, a server using a defined Network Control Program (NCP); See section 2.7.1 on page 8 for further details.

Messages may be excluded or included based on various predicates for viewing in the Workspace by defining a set of filters.

1.3 Message Types in the GUI

1.3.1 Apparent types

All messages 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.

1.3.2 Set types

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

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

See sub-section [4.12](#) on page [17](#) for details on how to assign the set type of a record.

1.4 summary

In summary, the NIP can be used to:

- View Network Messages in a hierarchical or tree form.
- Edit and change the content of a message.
- Create a new message from an existing message type.
- Use an Open Spec to read a Candidate Log File containing a log of messages.
- Use a Network Control Program to send and receive messages over a network.

2 Elements of the NIP

2.1 Overview

Various elements make up the hierarchical structure within the NIP Workspace.

This section explains the purpose of these elements, and how they are related. Each element is further explained starting at the technical reference; Section 3 on page 9.

The main elements are:

- The Workspace. Explained in section 2.2 on page 5.
- The Environment Variables collection. Explained in section 2.3 on page 5.
- The Message Types collection. Explained in section 2.4 on page 6.
- The Query collection. Explained in section 2.5 on page 6.
- The Candidate log file(s). Explained in section 2.6 on page 7.
- The Worksheet collection. Explained in section 2.7 on page 7.

2.2 The Workspace

The Workspace is the root of the hierarchy displayed in the NIP GUI. All of the collections are contained within the Workspace. Each Workspace is described in a single text based file with a specific grammar to describe the contents of the Workspace. A Workspace can be saved and later re-opened.

As well as the collections, the Workspace contains the following elements:

- Description

This is a free form text string used to describe the Workspace. It is often used to identify the purpose of the Workspace.
- The Object Types Definition File

This element identifies the Object Types Definition File which specifies the Object Type Definitions, restrictions and typing information of messages in the Workspace.

New messages can only be created from Object Types specified in this file.

2.3 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.

2.4 The Message Types collection

The message types collection element contains a list of all of the messages contained in the Object Types Definition File. These message types are displayed in a tree view which makes it easy to determine which fields a particular message contains. The message types in this section can be used to create a new message within a Worksheet.

2.5 The Query Collection

The query collection contains a list of all of the queries stored in the Workspace. Each query is an element itself, and has its own behaviour depending on the type of query.

A query element has a name, which is simply a name the user assigns to a query.

A query also has sub-elements which depend on the type of query it is. There are three types of queries. They are:

1. Message Type Query. Used to include or exclude the given message types.
2. Like ASCII/EBCDIC Query. Used to include or exclude messages that match a given regular expression.
3. Object Types Query. Used to include or exclude messages that match the specified Object Types query.

When a query is created certain operations (depending on the query type) can be performed over the data in the Candidate Log File.

The following sub sections will describe in more detail the different types of queries. For examples of using these queries refer to

2.5.1 Message Type Query

The Message Type query includes or excludes messages from a Candidate Log File. First the operation, include or exclude, is selected and secondly the message types from the Object Type Definitions are selected.

2.5.2 Like ASCII/EBCDIC Query

This query will filter all messages that match a regular expression. The regular expression can be either ASCII or EBCDIC based. The messages that match the regular expression can be either excluded or included from the view.

2.5.3 Object Types Query

This query allows for the use of a real Query using the Object Types query language. First the base message type to perform the query on is selected using a right-click on the Message Type element, and then the query can be entered in the WHERE clause value.

Substitutions in place of field names can be made by specifying wild cards, which are preceded by a dollar sign. When a substitution is used more nodes are created below the query element. A field name can then be specified by right clicking on the substitution name and selecting it from the list that appears.

2.6 The Candidate Log File(s)

The Candidate Log File element is used to identify a physical file containing a set of messages of the types specified in the Object Types Definition File element and the name of an Open Spec object that will interpret and load the messages into the hierarchical view. The Candidate Log File element contains a message collection element under which the loaded messages will be displayed.

The Open Spec object identifies a program DLL and any required parameters. Its purpose is the interpretation and loading of the messages from the Candidate Log File.

2.7 The Worksheet Collection

The Worksheet collection contains a list of all of the Worksheets used in the Workspace. Each of these Worksheets can have their own NCP defined.

Each Worksheet contains a message collection which is used to store a list of messages of interest and as an area where the values of fields can be altered and saved.

The Worksheet can also be recorded to a script if it contains an NCP and later re-run under the Eresia Test Environment.

Other functions of the Worksheet are:

- Storing a list of messages.
- Creating new messages using the message types collection.
- Creating copies of messages using the Candidate Log File.
- Sending messages over a network using an NCP.
- Recording a script from the Eresia Test Environment.

A new Worksheet can be added to the Worksheet collection at any time. Saving the Workspace saves all the information recorded in a Worksheet.

2.7.1 The Network Control Program

A Network Control Program element refers to a program DLL. Once the program is identified and loaded, parameter options are presented and can be modified. Examples of parameters are the network address and port of the server.

The NCP also has a set of properties that are used after the connection has been established, but before the message has been sent and received from the server or target.

The NCP element can be specified at either a Worksheet collection level or within an individual Worksheet. The Worksheet collection NCP acts as a default NCP definition.

2.7.2 Viewing Message entries

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

2.7.3 Editing and changing Message buffers

Messages can be edited by double-clicking the value of a field and entering the new value.

Once a Worksheet has been created messages can be created from the message types collection or copied from a Candidate Log File or from another message in any Worksheet.

Examples of the usage of messages is explained in detail in section [3](#) on page [9](#).

2.7.4 Sending Messages over a network

Once the NCP has been defined, the user can send and receive the response of individual messages within the Worksheet.

The response can be viewed in the hierarchy and, as any other message, be used to create new messages.

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 project, which are the Description, Message Types Collection, Query Collection, Worksheet Collection and Candidate Log 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 current Workspace, the user can add a description to the Workspace by right-clicking on the Workspace element and selecting 'Add' and then 'Description'. A new entry 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

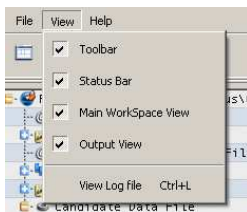
This section explains the layout of the menu bar. Individual menu items are also explained.

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

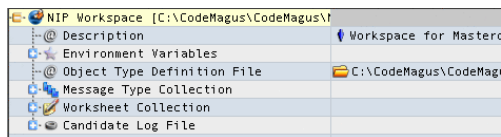


Actions	
Contents and Index	Display the help file for the NIP
About Eresia NIP	Show the Help/About dialog

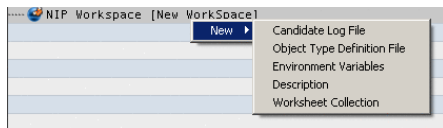
4 GUI Elements

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 NIP Workspace



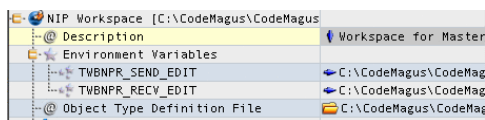
The NIP Workspace element contains all of the other elements pertaining to a project. It, and all the items below it, can be saved to a file; normally with a suffix of “.npw”.



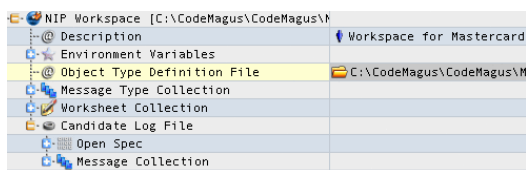
Actions	
New/Candidate Log File	Add a log file containing messages to the Workspace
New/Object Type Definition File	Add an Object Type Definition File
New/Description	Add a description of the Workspace
New/Query Collection	Add collection of queries to the Workspace
New/Worksheet Collection	Add collection of Worksheets to the Workspace
View Object Type Definition File	Display the OTDF in a window

4.2 Description

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

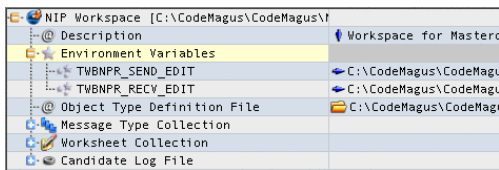


4.3 Object Types Definition File



The Object Types Definition File specifies the file which acts as the source of the definition of the message types which will be used in the Workspace. When an object types file is defined, an entry will appear under the Workspace which is called Message Types Collection. See section 4.6 on page 13.

4.4 Environment Variables collection

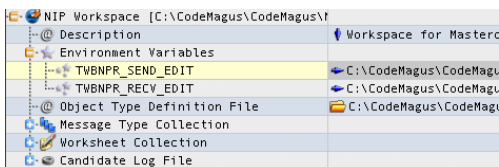


The Environment Variables Collection contains a list of all defined environment variables. Each of these has its own entry. See section 4.5 on page 12.

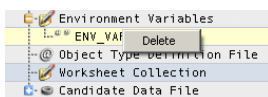


Actions	
New/Environment Variable	Add a new entry
Refresh	Refresh the list of variables
Delete	Remove this element

4.5 Environment Variable

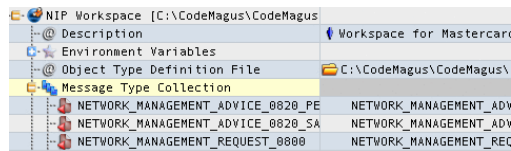


The Environment Variable node has a name and a value. The name is the actual name of the environment variable to set, and the value is the actual value.



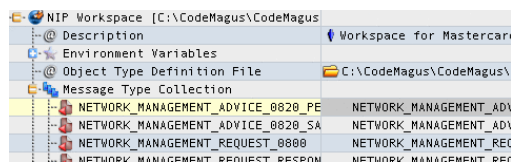
Actions	
Delete	Remove this element

4.6 Message Types Collection

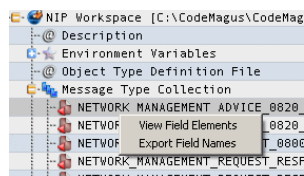


The Message Types Collection contains a list of all defined message types within the specified Object Types Definition File. Each of these message type entries can be used to get more information about individual types defined in the object types definition file. See section 4.7 on page 13

4.7 Message Type

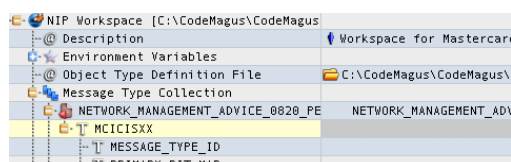


A Message Type element has the type's name, and can be used to view a full message of that type. These message types can be used to create a new, empty message if a type is dragged to a Worksheet, or if a user performs a Copy/Paste of a message type into a Worksheet.



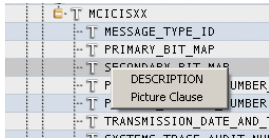
Actions	
View Field Elements	Populate the fields of the message in the tree
Export Field Names	Display a list of field names of the message

4.8 Message Type Field



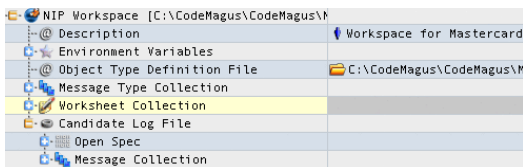
Each field within a message type displays the name of the field as it is defined in the object type and copy book. This field shows the name of the field. The value of this entry will always be empty since this element has no buffer. It serves only as a means

for the user to view which fields and sub-fields are contained within a given type. See section 4.13 on page 17 for an example of an editable field.



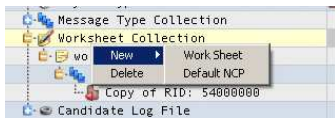
Actions	
DESCRIPTION	Display the description of the field as per the copybook
Picture Clause	Display the picture clause of the field as per the copybook

4.9 Worksheet Collection



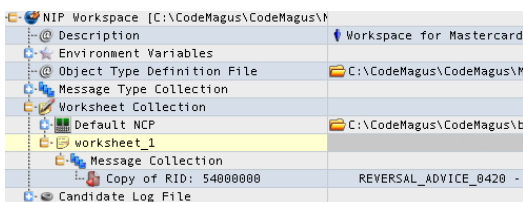
A Worksheet Collection contains a list of Worksheet elements (see section 4.10 on page 14).

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



Actions	
New/Work Sheet	Add a new Worksheet to this Worksheet Collection
New/Default NCP	Add a default NCP that applies to all Worksheets
Delete	Delete the Worksheet Collection

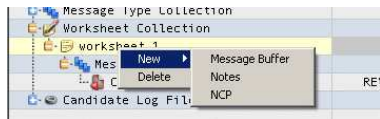
4.10 Worksheet



Worksheets are used to store a collection of records which are copied from a Candidate Log File. These records are physically stored in the workspace file when the workspace

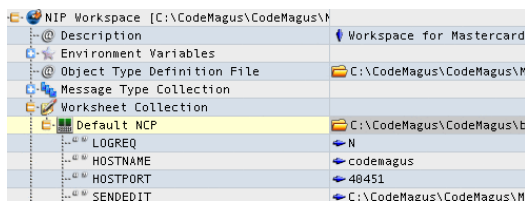
is saved.

Worksheets can be recorded if an NCP has been defined for the worksheet, or if a Default NCP has been defined in the Worksheet Collection.



Actions	
New/Message Buffer	Add an empty record buffer to the Worksheet
New/Notes	Add a text note to the Worksheet
New/NCP	Set the NCP for this Worksheet
Delete	Remove this Worksheet from the Worksheet Collection
Record	Record this Worksheet in the Automated Test Environment

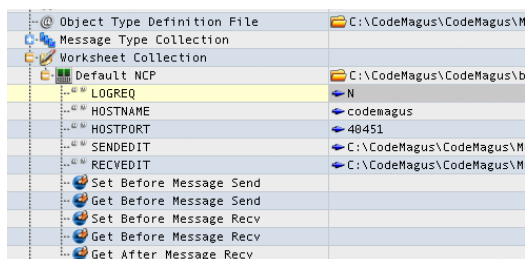
4.11 Network Control Program



The Network Control Program (NCP) allows the user to specify the parameters and the attributes used in the NCP to connect to a remote point. Parameters are used to specify information about the connection. Attributes are used to specify information of interest before and after the sending and receiving operations of the NCP.

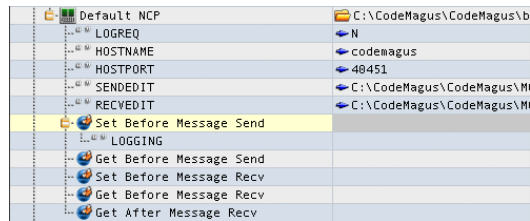
Both the parameters and the attributes are specified in the DLL, and are automatically populated in the GUI.

4.11.1 NCP Parameter



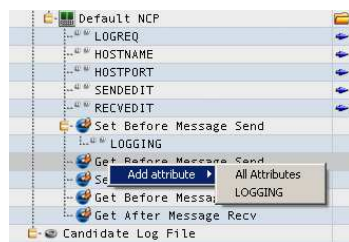
The required and optional parameters are simply populated in a list. These elements cannot be added or removed as this is done automatically by retrieving information from the NCP DLL itself.

4.11.2 NCP Attribute Group



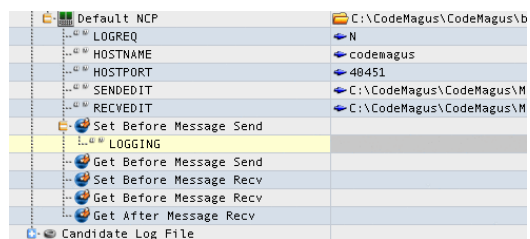
The Attributes area displays a list of all of the attributes used during the usage of the NCP. These parameters allow the user to specify which attributes are set and retrieved before and after the send and receive operations. These set and get operations are performed after the NCP network circuit is opened.

The user can specify which attributes are to be set and retrieved by right-clicking on the appropriate node and selecting it from the popup list. The user can also select all attributes. To delete an entry, the user can simply right-click on the attribute and select delete.

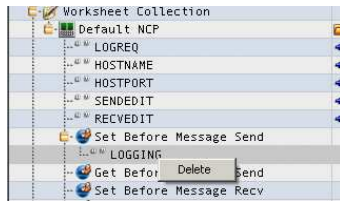


Actions	
Add attribute/All Attributes	Add all attributes to the node
Any other entry	Add only this attribute to the node

4.11.3 NCP Attribute

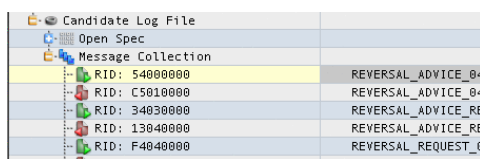


This node is simply one instance of an attribute in one of the NCP Attribute Groups. The menu here simply allows the user to delete this entry from its parent node.

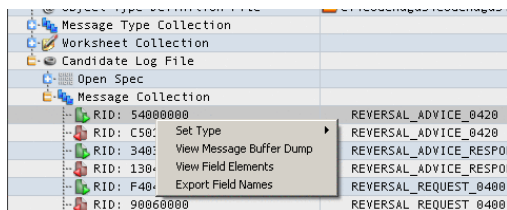


Actions	
Delete	Remove this message from its parent node

4.12 Message



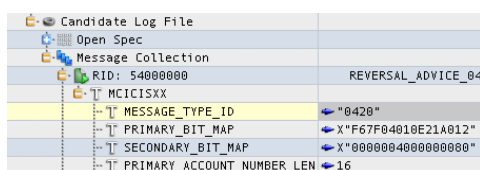
The Message element is a representation of a buffer of data which conforms to a type defined in the object types. This element has different actions available depending on its position in the tree.



Actions	
Set Type	Assign the Set Type of a record to a specific message type
Send Using NCP	Send this message over the network
View Message Buffer Dump	View the actual buffer of the message
View Field Elements	Populate the fields of the message in the tree
Export Field Names	Display a list of field names of the message
Delete	Remove this message from its parent Message Collection

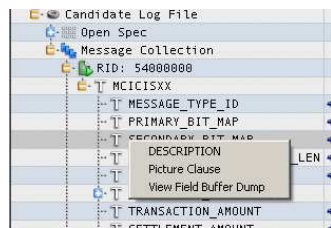
Note that only some of these actions will be available to the user depending on where this message is in the tree and on other conditions.

4.13 Message Field



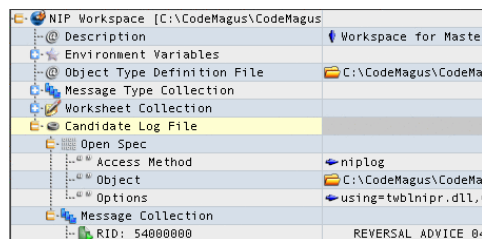
Each field within a message displays the name of the field as it is defined in the object type and copy book. This field name can not be changed, but the value of the data within this field can be changed if and only if this field is contained within a message which is in a Worksheet. This allows the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

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

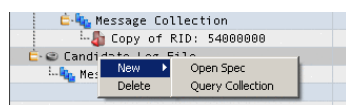


Actions	
DESCRIPTION	Display the description of the field as per the copybook
Picture Clause	View the picture clause of this field
View Field Buffer Dump	View the buffer dump of this field

4.14 Candidate Log File



A Candidate Log File (CLF) is a data file containing a number of messages, in a specific format. This file is read using an Open Spec object.



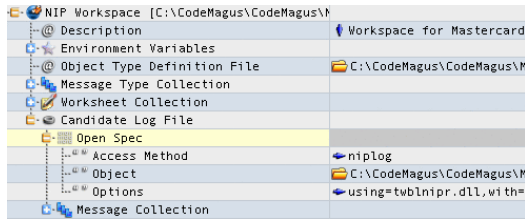
Actions	
New/Open Spec	Add an Open Spec object
New/Query Collection	Add a set of queries

The Open Spec under the Candidate Log 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 4.15 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 4.12 on page 17).

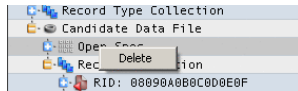
The Candidate Log File can also contain a Query Collection (see section 4.16 on page 20), which contains a list of queries that can be executed on this Candidate Log File.

4.15 Open Spec



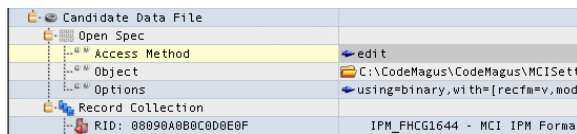
This element simply allows the user to set up the parameters to access the Candidate Log File (see section 4.14 on page 18).

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.



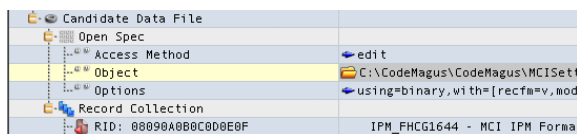
Actions	
Delete	Remove this Open Spec from its parent

4.15.1 Access Method



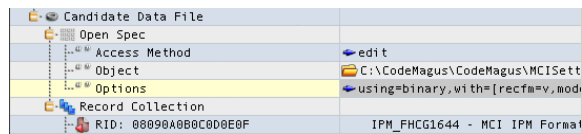
This element is used to specify which access method to use to read from or write to a data file (log). The name supplied here specifies the name of the AMD file to use to look up the required information. Some options for this item are FILE, BINARY and NIPLOG.

4.15.2 Object



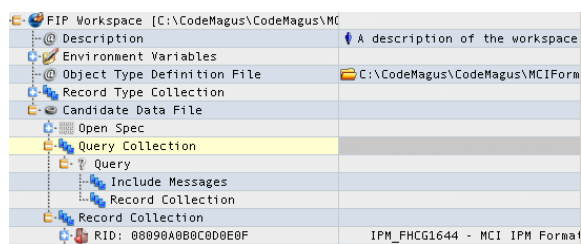
This element is used to specify the full path of the Object which the Open Spec will access using the specified Access Method.

4.15.3 Options



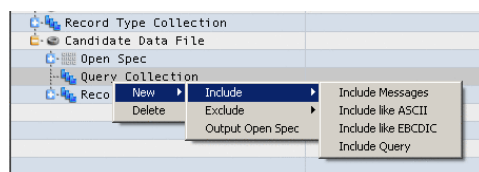
This element is used to specify the full set of options passed to the Open Spec when it is being opened for read or write access.

4.16 Query Collection



This element simply allows the user to add queries that can be performed on the Candidate Log File. See sections subsection 4.17.1 on page 21, subsection 4.17.2 on page 22 and subsection 4.17.3 on page 22 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/Messages	Add a message 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/Messages	Add a message 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
Delete	Remove this query collection from its parent

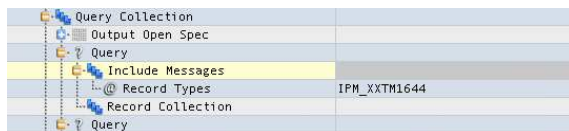
4.17 Queries

All Queries have the following menu options at their root node:



Actions	
Run Query	Execute the query
Delete	Remove this query from the query collection

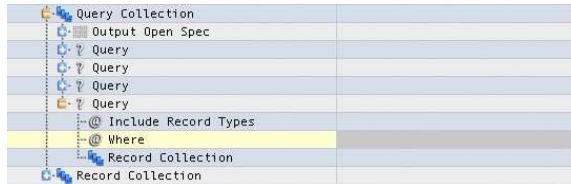
4.17.1 Include/Exclude Messages Query



This Query allows the user to filter the messages in a Candidate Log File by the message type. The user can either include or exclude a list of message types. The user can add to this list by right-clicking on the node 'Message Types' and selecting the message type from the list.

Actions	
Run Query	Perform the query and populate the message list
Delete	Remove this query from the query collection

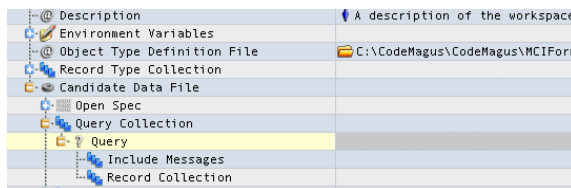
4.17.2 Include/Exclude ASCII/EBCDIC Query



This Query allows the user to filter the messages by performing a regular expression on the buffer of the message. 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 message list
Delete	Remove this query from the query collection

4.17.3 Include/Exclude Query



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

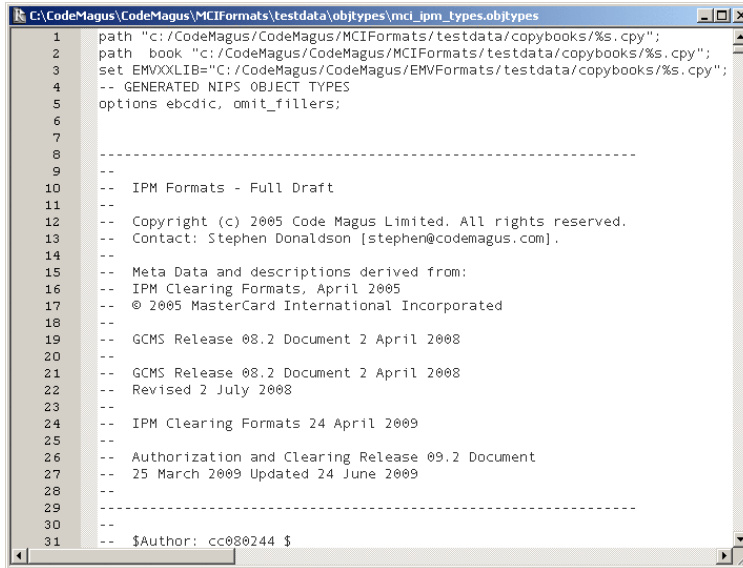
If the user specifies any substitutions in the search string, another node will appear under Substitutions. The user can then right click on each substitution and select the field name from the list.

Actions	
Run Query	Perform the query and populate the message list
Delete	Remove this query from the query collection

5 GUI Output Windows

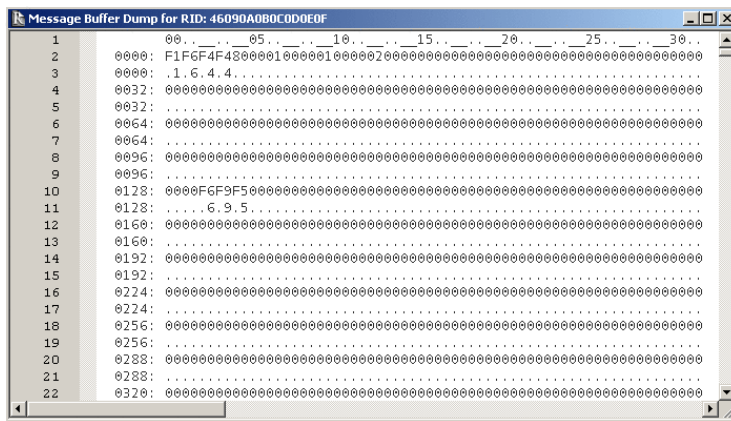
5.1 Object Type Definition File

This window displays the Object Type Definition File.



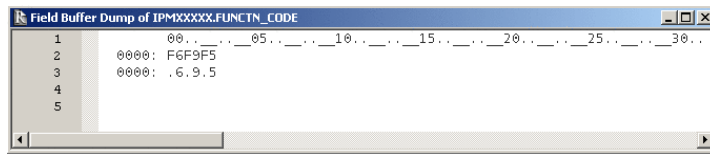
5.2 Message Buffer Dump

This window displays a the buffer of a given message.



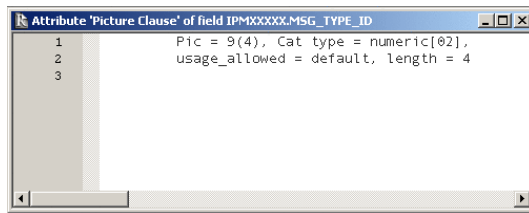
5.3 Field Buffer Dump

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



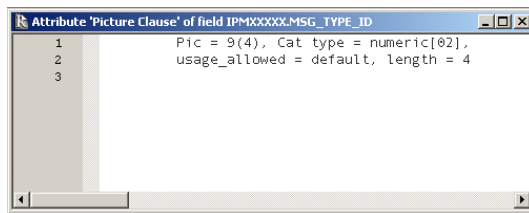
5.4 Field Picture Clause

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.

