CA DataMinder

External Socket Agent API Guide

14.6

CA Technologies Product References

This document references the following CA Technologies products:

- CA DataMinder™
- Windows, Microsoft and Active Directory are trademarks or registered trademarks of Microsoft Corporation.
- All other marks are owned by their respective companies.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- External Agent Socket API Header (see page 13)—Corrected data type of Priority.
- WGN_MSGSCHEMATYPE (see page 15)—Corrected schema ID.
- Sample Messages (see page 37)—Corrected position of Process Result field in two header examples.
- Clarified that all UINT4 types are big-endian (DWORDs).

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: About the External Socket Agent API 7

Audience 7

Chapter 2: Socket Agent 9

How the Socket Agent Works 10
Policy Engine Hub 11
External Agent Socket API Messages 11
- Flow of Messages 11
- Communications Example 12
- Message Description 13
- External Agent Socket API Request Message 19
- External Agent Socket API Response Message 23
- External Agent Socket API Completion Message 24
- External Agent Socket API Cancel Message 26

External Agent API Interface 27
- Import Object Type 27

Deferred Completion Processing 30
- IWgnImportConnectorDeferredCompletion 31

Registry Entries - Socket Agent 31
- AgentPort 31

Performance Monitor Counters 35
Installation 36
- Program Location 36
- Registry Entries 36

Appendix A: Sample Messages 37

External Agent Request Message 37
External Agent Response Message 39
External Agent Completion Message 40
External Agent Response Message (Completion) 41
External Agent Cancel Message 42

Chapter 1: About the External Socket Agent API

This document provides information on the Socket Agent and the External Socket API messages processed by the Socket Agent:

- The messages sent to the Socket Agent, for example, to request a File Smart Tag.
- The messages received from the Socket Agent, for example, containing the External Agent response.
- The calls made to, and the data passed to the External Agent API.
- The data received from the External Agent.

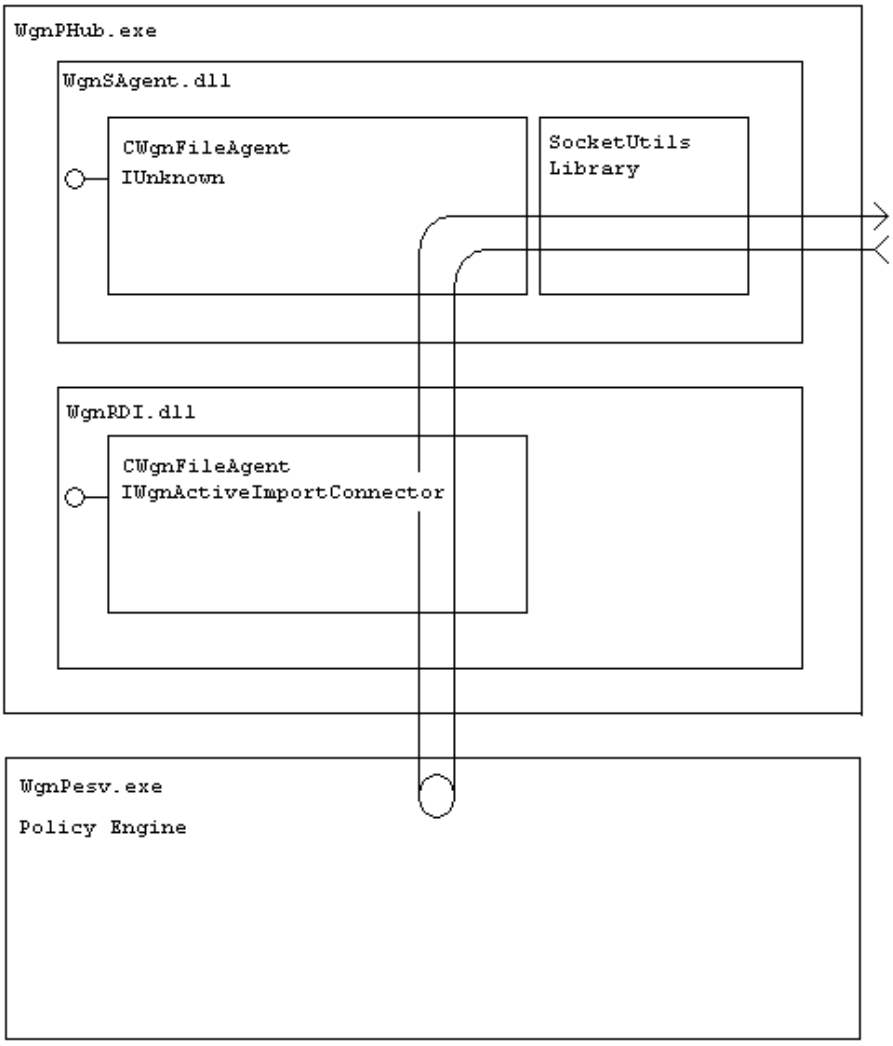
Audience

This document is written for Architects and Software Engineers who are interested in the operation of the External Agent Socket API.

Chapter 2: Socket Agent

The Socket Agent processes External Agent requests, for example, returning Smart Tags for emails or files. The service is accessed as a TCP/IP Server, with requests being passed in a format described in this document. The agent receives these requests, disassembles the components, and calls the External Agent API to process the request. The response from the External Agent API is packaged into the response message and returned to the caller.

Example: Socket Agent Configuration



If deferred completion is requested, and the message is processed successfully, the API requests that the application must send an External Agent completion message. The message indicates whether the event is to be committed or rolled back. The outcome of the processing completion message is returned to the application by another External Agent response message.

This section contains the following topics:

How the Socket Agent Works (see page 10)
Policy Engine Hub (see page 11)
External Agent Socket API Messages (see page 11)
External Agent API Interface (see page 27)
Deferred Completion Processing (see page 30)
Registry Entries - Socket Agent (see page 31)
Performance Monitor Counters (see page 35)
Installation (see page 36)

More information:

External Agent Socket API Messages (see page 11)

How the Socket Agent Works

The Socket Agent:

1. Socket Agent processes messages asynchronously.
2. The Socket Agent runs under the Policy Engine Hub. The agent is, however self-contained, enabling it to run also, if necessary, under the Policy Engine.
3. The Application passes the request to, and receives responses from the Socket Agent in the External Agent API Message format.
4. Applications send messages to the Socket Agent using TCP/IP. The Socket Agent acts as a TCPIP Server. The application may be running on a different Operating System, such as UNIX. The Socket Agent uses the Wgn common receiver classes. The Socket Agent always sends the response back on the same TCP/IP port that it was received on.
5. External Agent Socket API requests are passed to, and responses received from, the External Agent API, when the entire message has been received.
6. Socket Agent communications can be configured using Registry Settings.
7. An External Agent Requestor is required to assist the user in formatting the External Agent messages. The External Agent Requestor is not covered by this document.

Policy Engine Hub

The Policy Engine Hub can host the Socket Agent.

External Agent Socket API Messages

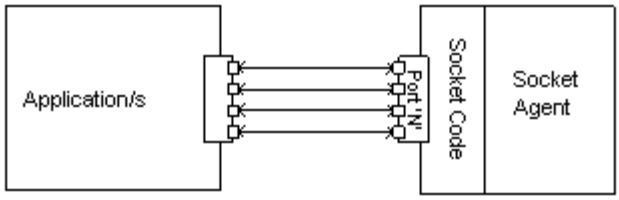
Flow of Messages

All communications with the Socket Agent are through TCP sockets. The following discussion and design assume the following:

Communications take place on a common configurable port number across all machines, but defaulting to 8538 (0x215A).

- There can be multiple simultaneous connections on each port (that is, multiple sockets).
- Communications on a single socket are bi-directional.
- Communication on each socket in each direction is in a continuous 'stream' of messages. The listener is responsible for splitting the stream into distinct messages.
- The caller is responsible for establishing socket connections to the Socket Agent, and maintaining the required number of sockets.
- Each message comprises a header and payload. The header size is fixed, whereas the payload can be arbitrarily long.
- Each message is self contained and atomic – the header identifies the type of message (for example, the command being executed) and all the required data to perform it.
- Authentication of the Socket Agent user is not deemed necessary as all these components are hosted on machines in proximity and existing LAN security protects all communication between them.

Example: TCP/IP Communications between the main components

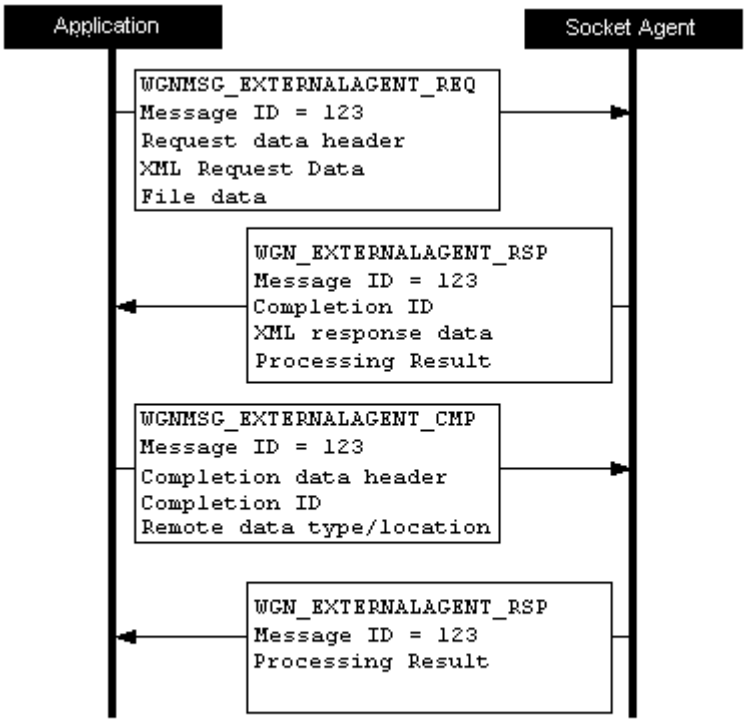


Communications Example

The following example illustrates the required functionality across the TCP interface.

All messages across the TCP interface contain a Message ID which identifies the mail being processed. In the example below, an application sends an External Agent request (WGNMSG_EXTERNALAGENT_REQ, with Message ID 123 to the Socket Agent.

Example: TCP communications for message processing



The Socket Agent processes the request and returns a message (WGNMSG_EXTERNALAGENT RSP) with the same Message ID containing an XML processing result. The External Agent Completion and the corresponding External Agent response only occur when deferred completion is requested.

A Socket Agent Response message is returned when the Socket Agent determines that the message contains invalid data. The full error message may be found in the Socket Agent log.

The conversation is terminated without a response if the Distributor or Socket Agent receives a message containing an invalid major schema. The full error message may be found in the Distributor log.

More information:

External Agent Socket API Messages (see page 11)

Message Description

External Agent Socket API Header

All External Agent Socket API messages contain a two part header plus a variable length payload, followed by footer. The overall length of the message **must** be an integer multiple of four bytes. Padding bytes (set to zero) may be required after each entry in the Payload field to achieve this.

The primary header contains the following fields:

Bytes	Field Name	Type	Description	Mandatory?
0 – 3	Schema ID	UINT4 Big-endian.	Identifies the schema for the message. This value must be set to WGNSCH_SKT_SERVER_VER1 as defined by the WGN_MSGSCHEMATYPE enumeration.	Yes
4 – 7	Message Size	UINT4 Big-endian.	Overall size of message including headers. This limits maximum file size to approx. 4 GB.	Yes

The format of the secondary header is as follows:

Bytes	Field Name	Type	Description	Mandatory?
0 – 3	RoutingID	UINT4 Big-endian	Can be used for load balancing with the Distributor. If used, it must be present in the Distributor Routing ID table.	No
4 – 7	Message ID	UINT4 Big-endian	Unique message identifier.	Yes
8 – 11	MessageType	UINT4 Big-endian	Identifies the message type - one of the WGN_MESSAGETYPE enumerations in Appendix B.	Yes
12	Priority	BYTE	Priority of the message 0 (low) 256 (high) (reserved for future use) – *Must be zero.* Zero means use the default priority.	Yes
13	Flags1	BYTE	FLAGS1_ExpectReply (0x01) Set by Socket Agent when a completion message is required.	-

14	Flags2	BYTE	FLAGS2_ReturnToSenderPort (0x01) FLAGS2_EnablePartialEvent (0x02) FLAGS2_MultipartMsgFirst (0x04) FLAGS2_MultipartMsgIntermediate(0x08) FLAGS2_MultipartMsgLast(0x0C)	-
15	Flags3	BYTE	Reserved – *Must be set to zero*	-
16-19	Process Result	HRESULT	*Must be set to zero for messages passed to the External Agent API.* On return, this field contains the HRESULT of processing the message.	No
20 – 23	Routing Context	UINT4 Big-endian.	*Internal use – The Distributor sets this to the IP address of the sender. The Distributor passes back the IP Address of the Socket Agent on the Response.* On an External Agent request message, this must be set to zero. When sending an External Agent completion message, the application sets these to the Routing Context from the original message response.	-
24 – 27	Reserved2	UINT4 Big-endian.	*Internal use – The Distributor sets this to the IP port of the sender.*	-
28 – 31	Reserved 3	UINT4 Big-endian.	*Reserved for internal use. Must be set to zero.*	-

Type Definitions

The following Type Definitions apply to the Secondary Header. These enumerated types contain three sub-fields, namely MAJORTYPE, MINORTYPE and SUBTYPE, where each sub field is defined by the following bit masks:

```
MAJORTYPEMASK = 0xFF000000
MINORTYPEMASK = 0x00FFF000
SUBTYPEMASK   = 0x00000FFF
```

WGN_MSGSCHEMATYPE

The WGN_MSGSCHEMATYPE enumeration identifies the schema to which the message conforms. This is required to allow backwards compatibility for future versions.

Note: This enumeration uses the MAJOR/MINOR/SUBTYPE schema outlined above.

```
typedef emum tagWGN_MSGSCHEMATYPE
{
     WGNSCH_SKT_SERVER_VER1        = 0x01001001
}
WGN_MSGSCHEMATYPE
```

WGN_MESSAGETYPE

The WGN_MESSAGETYPE enumeration indicates the type of the TCP message.

Note: This enumeration uses the MAJOR/MINOR/SUBTYPE schema outlined above.

```
typedef emum tagWGN_MESSAGETYPE
{
        WGNMSG_EXTERNALAGENT_REQ = 0x10001000,
        WGNMSG_EXTERNALAGENT_RSP = 0x10002000,
        WGNMSG_EXTERNALAGENT_CMP = 0x10003000,
        WGNMSG_EXTERNALAGENT_CAN = 0x10004000,
        WGNMSG_DISTRIBUTOR_RSP   = 0x01FFF000,
        WGNMSG_SOCKETAGENT_RSP   = 0x10FFF000
}
WGN_MESSAGETYPE
```

Selected Process Results

Possible process results include:

0x25631313

The message has successfully completed Event processing.

0x25631356

The policy engine has successfully completed event processing and is waiting for the caller to commit or abort - The message has been partially processed. A completion message is required to complete the processing.

Message Flags

These are used when running with the Distributor. Although these flags are used to control the Distributor, we recommend using them even if you are not using the Distributor. This ensures that the Distributor can be added later without change.

FLAGS2_ReturnToSenderPort (0x01)

Return the Response Message on the same port as the Request Message.

Set this flag to route the Response Message *only* to the original port that received the Request Message (or Completion Message). The Distributor always attempts to send the Response Message to the original port. If this flag is not specified, and the Distributor cannot send the Response Message to the original port, then it sends the Response Message to any open port on the same machine.

FLAGS2_EnablePartialEvent(0x02)

Enable partial events.

Set this flag to enable partial event processing for this request. The Socket Agent requests the External Agent API to create a partial event containing only the XML Metadata.

Note: This flag is allowed, but ignored in the current version.

FLAGS2_MultipartMsgFirst (0x04)

FLAGS2_MultipartMsgIntermediate(0x08)

FLAGS2_MultipartMsgLast(0x0C)

Use these flags when the message data is split over multiple messages. Multipart messages are recommended if the total message size would exceed 90 MB – use multipart messages to send the message in messages of less than 90 MB. These flags can be is used with the Data Completion flag of a Data Element – if the data is incomplete, the message must be incomplete, and vice versa. Multipart messages are allowed for Request and completion messages only.

External Agent Socket API Footer

Each message has a footer of 4 bytes.

MessageSize-4 - MessageSize-1	End of message Signature	UINT4 Big-endian	Fixed bit pattern to confirm that the message is of the correct size (0x99999999)	Yes

Message Payload

All messages contain a data header, which can be followed by 0 or more Data Elements. The format of the message header is message specific. Data Elements have a generic format, but the data format of data elements are message specific.

Data Elements

Bytes	Field Name	Type	Description	Mandatory?
1 – 4	Data Size	UINT4 Big-endian.	The size of this data element, including the Data Size, Data Type and Data, but excluding any padding of the data.	Yes
5 – 6	Data Flags	UINT2 Big-endian.	0x01 – FLAGSDATA_Incomplete	Yes
7 – 8	Data Type	UINT2 Big-endian.	Must be from the WGN_IMPORTDATA_TYPE enumeration:	Yes
9 – x	Data	Bytes	An array of bytes containing the data for this data element. The data must be padded with NULLS to a multiple of 4 bytes.	No
x+1 - End of data	Padding	BYTE=0	Null padding of 0-3 bytes to ensure that the end of the data element is 4 byte aligned	Yes

Data Flags

FLAGSDATA_Incomplete

Use this flag when you want to continue a data element across multipart messages. This flag must be used with the Multipart Message flags on the message secondary header.

Data Type

The Data Type enumeration, defined in SocketUtils.h, contains the following items:

```
WGN_IMPORTDATA_TYPE_DATA                = 1
WGN_IMPORTDATA_TYPE_MESSAGEID           = 2
WGN_IMPORTDATA_TYPE_STREAMNAME          = 3
WGN_IMPORTDATA_TYPE_XML_METADATA        = 4
WGN_IMPORTDATA_TYPE_XML_RESULTS         = 5
WGN_IMPORTDATA_TYPE_HRESULT_TEXT        = 6
WGN_IMPORTDATA_TYPE_XML_ADD_ATTRIBUTES = 7
```

For Object Type MAPI, WGN_IMPORTDATA_TYPE_DATA, the data contains the data from a Outlook Message Format (.msg) file.

For Data Format WGN_IMPORTDATA_TYPE_DATA , the data contains valid RFC 822 mail data. For example, the contents of an Outlook Express Mail (.eml) file.

For Data Format WGN_IMPORTDATA_TYPE_DATA , the data contains the contents of the file, as it is stored on disk. If, for example, the file is an IStorage file, the message may contain multiple streams, there will be one data element for each stream in that file.

For all other data formats, the data must be in UTF_16 big endian format.

Message ID Data Element

The Message ID data is defined as follows:

Bytes	Field Name	Type	Description	Mandatory?
1 - 4	Input Source	UINT4 Big-endian.	A unique application identifier supplied by CA Technologies.	Yes
5 – 8	Message ID Size	UINT4 Big-endian.	The size of this message id, in bytes.	Yes
13 – x	Message ID	Bytes	An array of bytes containing the Message ID.	No

This must be sent within a data element of type WGN_IMPORTDATA_TYPE_MESSAGEID.

The Message ID data must be a UTF-16 string in Network Byte Order (big endian order), up to 507 UTF_16 characters long.

The Message ID string is a string that uniquely identifies the email or file.

External Agent Socket API Request Message

The External Agent Request message consists of the standard message headers and footer, enclosing an External Agent Request payload. The External Agent Request payload can include, in order, a request data header (required), a Message ID data element, an XML metadata data element (both optional), and zero or more file / email data elements. Each External Agent Request Data Element must be padded to 4 bytes with NULLs.

Example: External Agent API Request Message

An External Agent API request message contains the following:

- Message Header
- Request Data Header
- Message ID Data Elements
- XML Metadata Element
- Data and Stream Data Elements
- Message Footer

An example External Agent Request Message may be found in the Appendix: External Agent Request Message.

More information:

Data and Stream Data Elements (see page 22)
Message ID Data Element (see page 18)
External Agent Request Message (see page 37)
External Agent Socket API Cancel Message (see page 26)
External Agent Socket API Footer (see page 16)
XML Metadata Element (see page 21)

External Agent Socket API Request Data Header

The External Agent Socket API Request Data Header is at the beginning of the data packet.

Bytes	Field Name	Type	Description	Mandatory?
1 – 4	Object Type	UINT4 Big-endian	A member of the enumeration: WGN_OBJECT_TYPE.	Yes
5 – 8	Object Format	UINT4 Big-endian	A member of the enumeration: WGN_OBJECT_FORMAT.	Yes

9 – 12	Object Transport	UINT4 Big-endian.	A member of the enumeration: WGN_OBJECT_TRANSPORT.		Yes
13 – 14	Request Flags	UINT2 Big-endian.	Enable Deferred Completion	0x0001	Yes
15 – 16	Response Flags	UINT2 Big-endian.	Return XML	0x0001	Yes
17 – 20	XML Results Flags	UINT4 Big-endian.	Selects XML Results that will be returned in the response. Zero indicates return the default data elements.		No
21 – 24	Element Count	UINT4 Big-endian	Number of Data Elements.		Yes
25 – 28	Reserved 1	UINT4 Big-endian.	Must be zeros.		Yes
29 – 32	Reserved 2	UINT4 Big-endian.	Must be zeros.		Yes

Object Types

The External Agent Socket API supports the following Object Types:

```
WGN_IMPORTOBJ_TYPE_EMAIL  = 0
WGN_IMPORTOBJ_TYPE_FILE   = 2
```

Object Format

The External Agent Socket API supports the following Object Formats:

```
WGN_IMPORTOBJ_FORMAT_MAPI     = 0
WGN_IMPORTOBJ_FORMAT_RFC822   = 1
WGN_IMPORTOBJ_FORMAT_FILE     = 4
```

Object Transport

The External Agent Socket API requires the data to be in the message, and that the Object Transport to be WGN_IMPORTOBJ_TRANSPORT_MEMORY.

```
WGN_IMPORTOBJ_TRANSPORT_MEMORY = 2
```

Only some combinations are currently available. See the External Agent API documentation for more information.

Request Flags

The Application may set the Enable Deferred Completion flag.

Response Flags

Flags used to identify which Data Elements the Response Message should contain. One or more flags may be OR-ed together to specify multiple elements. The Response Header Data is always returned. The HRESULT Text data element will always be returned if the processing result is non-zero.

```
WGN_IMPORTOBJ_RETURN_XML = 0x0001
```

External Agent Socket API Request Data Elements

On the External Agent Request, the first data element, if present, must be the Message ID, followed by the XML metadata (if present), followed by 1 or more data elements containing the email or file data.

The *Message ID* data element consists of Message ID data contained in a data element having a data item type WGN_IMPORTOBJ_TYPE MESSAGEID. The message id data element must be NULL padded to a multiple of 4 bytes. The data element size includes the length of the data element size, the data element type and the data. It does not include the NULL padding. The Message ID data element is mandatory, although the Message ID can be zero length.

More information:

Data Elements (see page 17)
Message ID Data Element (see page 18)

XML Metadata Element

The *XML metadata* data element has a data item type WGN_IMPORTOBJ_TYPE_XML_METADATA. The XML metadata data must be a UTF-16 string in Network Byte Order (big endian order). The XML metadata must be NULL padded to a multiple of 4 bytes. The data element size includes the length of the data element size, the data element type and the data. It does not include the NULL padding.

If more than 1 XML Metadata element is contained in a message, the XML contained in all the XML Metadata elements is merged before the External Agent API is called.

Data and Stream Data Elements

The contents of the data elements depends on the value of the object type defined in the Request Data Header. Mail Object Types contain a single data element containing the email data. File Objects should always contain a primary data stream, contained in single data element containing the file data. They may also contain addition stream data, each stream contained in two data elements, the first data element containing stream name, the second data element containing the stream data. In all cases, there must be at least 1 Data Element of type WGN_IMPORTOBJ_FORMAT_DATA. If the file or email is empty, the data element will be empty.

MAPI Object Type

The data element must contain valid Mail Message data, in the same format as an Outlook Message file. The data element data type should be set to WGN_IMPORTOBJ_TYPE_DATA.

RFC822 Object Type

The data element must contain valid Mail Message data, in the in RFC2822 format. The data element data type should be set to WGN_IMPORTOBJ_TYPE_DATA. For example, the contents of an Outlook Express Mail (.eml) file.

File Object Type

The data element must contain the file data. The data element data type should be set to WGN_IMPORTOBJ_TYPE_DATA.

Additional file streams are defined by pairs of data elements, the first, which is optional, contains the stream name. The second data element contains the stream data. The stream name data element (data type WGN_IMPORTOBJ_TYPE_STREAMNAME), if present, must be specify the name in UTF_16 big endian characters. The data element containing the file stream data (data type WGN_IMPORTOBJ_FORMAT_DATA).

More information:

External Agent Socket API Request Data Header (see page 19)

External Agent Socket API Response Message

The External Agent Socket API Response payload can include, in order, a response data header (always present), an HRESULT text data element, and an XML result data element. The XML result data is discussed in the External Agent API. Each Response Data Element is NULL padded to a multiple of 4 bytes. The exact data elements returned depends on the *Return Data Elements* field of the External Agent Socket API Request Message.

Example: External Agent API Response Message

An External Agent API response message contains the following:

- Message Header
- Response Data Header
- HRESULT Text Data
- XML Result Data
- Message Footer

Example External Agent Response Messages may be found in the Appendix 1: External Agent Response Message and External Agent Response Message (Completion).

More information:

External Agent Response Data Header (see page 23)
External Agent Response Message (see page 39)
External Agent Response Message (Completion) (see page 41)
External Agent Socket API Cancel Message (see page 26)
External Agent Socket API Footer (see page 16)
HResult Data Element (see page 24)
XML Result Data Element (see page 24)

External Agent Response Data Header

The External Agent Socket API Response Data Header is at the beginning of the data packet.

Bytes	Field Name	Type	Description	Mandatory?
1 – 4	Completion ID	UINT4 Big-endian.	A non-zero number, returned by the External Agent that is used to uniquely identify the message.	Yes
5-8	Process Result	HRESULT	This field contains the HRESULT of processing the message. This is the same process result as may be found on the secondary header.	Yes

9 – 12	Element Count	UINT4 Big-endian	Number of Data Elements.	Yes
13 – 16	Reserved 1	UINT4 Big-endian.	Reserved.	No

External Agent Socket API Response Data Elements

The response data may include HRESULT and XML Result Data Elements.

More information:

Data Elements (see page 17)

HResult Data Element

The HRESULT text data element (Object Type WGN_IMPORTDATA_TYPE_HRESULT) contains the text of the processing HRESULT, in UTF_16 big endian characters. This data element is only present if the processing HRESULT is non-zero.

XML Result Data Element

The XML Result data element (Object Type WGN_IMPORTDATA_TYPE_XML_RESULTS) – if one is returned by the External Agent API, and the XML Result is selected in the External Agent Request Response Flag as page.

External Agent Socket API Completion Message

When building the Completion Message, the Application should set the Secondary Header Routing Context from the Response Message Secondary Header.

If an incomplete message does not receive a completion message within 1 minute, the incomplete message is timed out and rolled back. No further message regarding this message ID is sent to the application.

More information:

IncompleteMsgTimeout (see page 33)

External Agent Socket API Completion Message Payload

The Completion Message payload can include, in order, a completion data header (required), and an additional XML metadata data element. Each External Agent Completion Data Element must be padded to 4 bytes with NULL characters.

Example: External Agent API Completion Message

An External Agent API completion message contains the following:

- Primary Message Header
- Secondary Message Header
- Completion Data Header
- Message ID Data Element
- Additional XML Metadata Data Element
- Message Footer

An example External Agent Completion Messages may be found in the Appendix 1: External Agent Completion Message.

More information:

External Agent Completion Message (see page 40)

External Agent Completion Data Header

The External Agent Completion Data Header is at the beginning of the data packet.

Bytes	Field Name	Type	Description	Mandatory?
1 – 4	Completion ID	UINT4 Big-endian.	A number, returned on the External Agent Socket API response message that is used to uniquely identify the message.	Yes
5 – 6	Completion Flags	UINT2 Big-endian	Do Commit 0x0001	Yes
7 – 8	Response Flags	UINT2 Big-endian	No Flags currently.	Yes
9 – 12	Element Count	UINT4 Big-endian.	Number of Data Elements.	Yes
13 – 16	Reserved1	UINT4 Big-endian.	Reserved. Must be zero.	Yes

If FLAGS1_DoCommit is not set, the completion is rolled back.

External Agent Completion Data Elements

The External Agent Completion may contain 0 or 1 XML attributes data elements.

The *Message ID* data element consists of Message ID data contained in a data element having a data item type WGN_IMPORTDATA_TYPE MESSAGEID. The message id data element must be NULL padded to a multiple of 4 bytes. The data element size includes the length of the data element size, the data element type and the data. It does not include the NULL padding. The Message ID data element is optional.

The *Additional XML Metadata data element* contains additional XML metadata in a UTF-16 big endian string. This is not currently implemented by the External Agent API. This data element has a data item type WGN_IMPORTDATA_TYPE_XML_METADATA.

More information:

Message ID Data Element (see page 18)

External Agent Socket API Cancel Message

The Message ID of the Cancel Message is the same as that of the Message being cancelled. No response is sent for the cancel message.

If the message is currently being processed by a Policy Engine, the message will be allowed to complete, however, the response will not be sent.

External Agent Socket API Cancel Message Payload

The Cancel Message contains no data header or data elements.

Example: External Agent API Cancel Message

An External Agent API cancel message contains the following:

- Primary Message Header
- Secondary Message Header
- Message footer

An example External Agent Cancel Messages may be found in the Appendix 1: External Agent Cancel Message.

More information:

External Agent Cancel Message (see page 42)

External Agent API Interface

The External Agent Socket API provides a TCP/IP Interface for the External Agent API - a COM Interface (Active Import Connector Interface) on the Wigan Active Import Connector object. On that Interface, the Import Object function may be used to apply policy to a given email or file.

```
[helpstring("method ImportObject")]
HRESULT ImportObject(
        [in]    WGN_IMPORTOBJ_TYPE      objectType,
        [in]    WGN_IMPORTOBJ_FORMAT    objectFormat,
        [in]    WGN_IMPORTOBJ_TRANSPORT objectTransport,
        [in]    VARIANT objectData,
        [in, unique]    LPCOLESTRLPOLESTR       messageId,      // OPTIONAL
        [in]    DWORD   importSource,
        [in, unique]    LPCOLESTR       xmlMessageAttributes,//OPTIONAL
        [in]    DWORD   eventSizeHintBytes,
        [in]    BOOL    allowDeferredCompletion,
        [out]   IWgnImportConnectorResult** results
        );
```

Import Object Type

Type

[in] WGN_IMPORTOBJ_TYPE_EMAIL, Required

Data

Set to one of the following existing values (dependant on the object type):

```
WGN_IMPORTOBJ_TYPE_EMAIL (0)
WGN_IMPORTOBJ_TYPE_FILE (2)
WGN_IMPORTOBJ_TYPE_PRECREATED (3)
```

Single stream files use WGN_IMPORTOBJ_TYPE_FILE, multi-stream files use

`WGN_IMPORTOBJ_TYPE_PRECREATED.`

Import Object Format

Type

[in] WGN_IMPORTOBJ_FORMAT, Required

Data

Set to one of: the following existing values (dependant on the object format):

```
WGN_IMPORTOBJ_FORMAT_MAPI        0x00
WGN_IMPORTOBJ_FORMAT_RFC822      0x01
WGN_IMPORTOBJ_FORMAT_FILE        0x04
WGN_IMPORTOBJ_FORMAT_PRECREATED 0x04
```

Import Object Transport

Type

[in] WGN_IMPORTOBJ_TRANSPORT, Required

Data

All data is converted to IStreams.

- For all files, the Socket Agent adds the IStreams to an Import File Object, and sets the Import Object Transport to WGN_IMPORTOBJ_TRANSPORT_COM.
- For Emails, the Import Object Transport is set to WGN_IMPORTOBJ_TRANSPORT_ISTREAM.

Object Data

Type

[in] VARIANT, Required

Data

This variant contains the appropriate Interface Pointer.

Message ID

Type

[in] LPOLESTR, Optional

Data

The message ID is a character string unique supplied by the Application.

xmlMessageAttibutes

Type

[in] LPOLESTR (XML Data), Optional

Data

The content of the XML Message Attributes is transparent to the Smart Tag Agent, and is included here as an example of some of the items that are included.

```
<?xml version="1.0" encoding="utf-16" ?>
<apm schema_version="1" xmlns=http://www.orchestria.com
    xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
    xsi:schemaLocation="http://www.orchestria.com xmleventattributes.xsd">
    <event>
            <file>
                    <host>ta-gf-wxptest</host>
                    [set the File Name variable]gf2warn.eml</filename>
                    <path>c:\test\data\smarttag</path>
            </file>
    </event>
</apm>
```

Event Size Hint Bytes

Type

[in] DWORD

Data

The total size of the data to be processed.

Allow Deferred Completion

Type

[in] BOOL

Data

Set to FALSE.

This flag is set on the Socket Agent packet data header – Enable Deferred Completion.

More information:

External Agent Socket API Request Data Header (see page 19)

Import Connector Result

Type

[out] IWgnImportConnectorResult**

Data

The location of a Pointer to an Importer Connector Result Interface.

The Import has 2 functions to extract the result:

HRESULT GetResultXML([out] LPOLESTR *pXMLVal);.

Used to retrieve the policy analysis results encoded in XML.

Example XML result

```
<?xml version="1.0" encoding="UTF-16" ?>
<apm schema_version="1" xmlns=http://www.orchestria.com
        xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
        xsi:schemaLocation="http://www.orchestria.com xmleventattributes.xsd">
        <policy>
                <state>
                <smart_tags>
                        <smart_tag name="My Smart Tag">
                                <value>A Smart Tag</value>
                        </smart_tag>
                </smart_tags>
        </state>
        </policy>
</apm>
```

Note: The Active Import Connector Interface has an additional method allowing Import Object asynchronously. This is currently not supported by the External Agent Socket API.

Deferred Completion Processing

This section describes the processing of the External Agent Socket API Completion Message.

More information:

External Agent Socket API Completion Message (see page 24)

IWgnImportConnectorDeferredCompletion

The call to the IWgnActiveImportConnector Import Object method returns an IWgnImportConnectorResult interface. We call the IWgnImportConnectorResult –Get Deferred Completion method to get and interface to IWgnImportConnectorDeferredCompletion.

If the Message ID element is supplied on the completion message, we call SetMessageID:

```
HRESULT SetMessageID(
        [in]    DWORD   importSource,
        [in]    LPCOLESTR       messageId ); //OPTIONAL
```

If the message requested Commit, we call the Commit method:

```
HRESULT Commit();
```

If the message requested Rollback, we call the Rollback method:

```
HRESULT Rollback();
```

Note: The Deferred Completion Processing Interface has an additional method allowing asynchronous commits. This is currently not supported by the External Agent Socket API.

Registry Entries - Socket Agent

The following registry entries control the operation of the Socket Agent. The Socket Agent does not check for changes to the registry. If you change the registry, you must restart the Policy Engine Hub to put those changes into effect.

Key Location

```
HKLM\SOFTWARE\CA\CA DataMinder\
    Active Policy Management\CurrentVersion\SocketAgent
```

AgentPort

Type

REG_DWORD

Data

This is the port number used for communication between the Application and the Socket Agent. This must be set to the same port number as used by the Application. The default is 8538 (0x215A).

CreateEAReq

Type

REG_DWORD

Data

Creates a diagnostic file containing the External Agent Socket API Request message as sent by the Application to the Socket Agent. The file name format is <message id>-<date>-<three digit ID>.EAReq Completion requests are also selected by this Registry Entry. Completion messages are saved with the File Format <message id>-<date>-<three digit ID>.EACmp.

This registry value can be set to:

0 Never - create the diagnostic file for any message (default)

1 Always - create a diagnostic file for all messages

2 On error - create a diagnostic file for any message that has an error

The file is only saved if the DiagnosticFolder registry parameter is also set.

CreateEARsp

Type

REG_DWORD

Data

Creates a diagnostic file containing the External Agent Socket API Response message as sent by the Socket Agent to the Application. The file name format is <message id>-<date>-<three digit ID>.EARsp.

This registry value can be set to:

0 Never - create the diagnostic file for any message (default)

1 Always - create a diagnostic file for all messages

2 On error - create a diagnostic file for any message that has an error

The file is only saved if the DiagnosticFolder registry parameter is also set.

DiagnosticFolder

Type

REG_SZ

Data

Specifies the Operating System folder used to store diagnostic files created using the CreateEAReq and CreateEAResp registry values. If this registry value is not specified, the Socket Agent will not create diagnostic files.

HubFailureMode

Type

REG_SZ

Data

Determines the action taken for new transactions when the Policy Engine Hub enters throttling mode.

Wait

Wait for a transaction that is currently processing to complete, before processing the new transaction.

Fail

Fail all new transactions until the Policy Engine Hub exits throttling mode.

IncompleteMsgTimeout

Type

REG_DWORD

Data

Defaults to 60000 milliseconds (1 minute). This is the amount of time the Socket Agent will hold an incomplete message before rolling back the completion. Messages are considered incomplete if a process request message with deferred completion has been received and processed, but no completion message has been received. Timed out messages are rolled back. Set the IncompleteMsgTimeout to zero to prevent messages will being timed out. The minimum IncompleteMsgTimeout is 5 seconds (5000 milliseconds).

Note: We recommend that you use the default timeout; specifying a short timeout or no timeout may adversely affect performance.

LogLevel

Type

REG_DWORD

Data

Determines the level of logging for Socket Agent activity. The Socket Agent can log errors, warnings, information or trace messages. Error might include incorrect request data. Supported logging levels are:

0 No log entries are written

1 Errors only

2 Errors and warnings (default)

3 Errors, warnings plus any extra information available

4 Errors, warnings, information entries, plus trace details

Note: You will need to restart the CA DataMinder Socket Agent service for changes to the logging level to take effect. Also, note that LogLevel=4 is provided for diagnostic purposes only.

LogMaxNumFiles

Type

REG_DWORD

Data

Defaults to 10. This specifies the maximum number of log files. When the maximum number of log files exists and the maximum size of the latest is reached (see above), the oldest log file is deleted to enable a new one to be created.

LogMaxSizeBytes

Type

REG_DWORD

Data

Defaults to 1,000,000. This specifies the maximum size for each log file. When the current log file reaches its maximum size, the Socket Agent creates a new log file. Log entries are written to the WgnSAgent_<date>.log file, where <date> indicates when the file was created. By default, log files are saved in the \System\Data\Log subfolder of the CA DataMinder installation folder on the server hosting the Socket Agent.

LoopbackHResult

Type

REG_DWORD

The LoopbackHResult registry setting is used to test communications between an Application and the Socket Agent. The Socket Agent responds to External Agent Requests without calling the External Agent API. The response contains the header and footer, with the Secondary Header Process Result being set to the given value. This behavior is triggered by the presence of this registry parameter, regardless of the registry value.

MaxMsgSize

Type

REG_DWORD

Data

Defaults to 100 MB. This is the maximum message size that is accepted by the Socket Agent. If a message exceeds this size, the socket is dropped. Set the MaxMsgSize to zero to disable this check.

Performance Monitor Counters

The Socket Agent Performance counters that are provided are as follows:

Counter	Description
Connections	The number of socket connections being managed by this communications link.
Forwarding Fails (Unknown destination)	The number of messages that were not sent because the destination was unknown. This counter does not apply to the Socket Agent.
Forwarding Fails (Unreachable destination)	The number of messages that were not sent because the destination was unreachable. This counter does not apply to the Socket Agent.
Incoming Messages	The number of messages received.
Incoming Messages/sec	The number of messages received per second.
Incoming Bytes	The number of bytes received.
Incoming Bytes/sec	The number of bytes received per second.
Messages failed (analysis)	Items that failed during analysis.
Messages failed (timeout)	The number of items failed by event timeout.
Messages failed (total)	The total number of items that have failed analysis.
Messages incomplete	Current Incomplete Items

Messages processed	The total messages processed.
Messages processing	The number of messages currently being processed.
Outgoing Messages	The number of messages successfully sent.
Outgoing Messages/sec	The number of messages successfully sent per second.
Outgoing Bytes	The number of bytes successfully sent.
Outgoing Bytes/sec	The number of bytes successfully sent per second.
Processing Time	Average processing time from when a complete message has been received, until the response is about to be sent.

When implementing throttling, new counters might include:

Counter	Description
Messages failed (throttling)	The number of items failed by throttling in 'Fail' mode - not applicable if in 'Wait' mode.

Installation

Program Location

The Socket Agent will be installed in the CA DataMinder Client directory.

Registry Entries

Installation Registry Entry Values

```
AgentPort               REG_DWORD   0x0000215A (8538)
DiagnosticFolder        REG_SZ
LogLevel                REG_DWORD   0x00000002 (2)         Errors and Warnings
LogMaxSizeBytes         REG_DWORD   0x000f4240 (1,000,000)
LogMaxNumFiles          REG_DWORD   0x0000000a (10)
CreateEAReq             REG_DWORD   0x00000002 (2)         Create on error
CreateEARsp             REG_DWORD   0x00000002 (2)         Create on error
IncompleteMsgTimeout    REG_DWORD   0x0000EA60 (60,000)    60 Seconds
MaxMsgSize              REG_DWORD   0x06400000 (100MB)
```

Appendix A: Sample Messages

This section contains the following topics:

External Agent Request Message (see page 37)
External Agent Response Message (see page 39)
External Agent Completion Message (see page 40)
External Agent Response Message (Completion) (see page 41)
External Agent Cancel Message (see page 42)

External Agent Request Message

This is a sample External Agent Request message sending file c:\test\data\ExtAgent\Small_File.txt (14 bytes).

```
External Agent Request ============================================
```

Header

```
Schema          : 0x01001001
Message Size    : 504
Routing ID      : 0
Message ID      : 1
Message type    : 0x10001000 (External Agent Request)
Priority        : 0
Flags1          : 0x00 (No Flags set)
Flags2          : 0x00 (No Flags set)
Flags3          : 0x00 (No Flags set)
Hr Process      : 0x00000000
Routing Context : 0x00000000
Reserved 2      : 0x00000000
Reserved 3      : 0x00000000
```

Message data

Data Header

```
Object Type      : 0x00000002 (File)
Object Format    : 0x00000004 (File)
Object Transport : 0x00000002 (Memory)
Request Flags    : 0x0001 (Enable Deferred Completion)
Response Flags   : 0x0001 (Return XML)
XML Results Flags: 0x00000001 (Smart Tags)
Element Count    : 0x00000003
Reserved 1       : 0x00000000
Reserved 2       : 0x00000000
```

Data Element (1)

```
  Data Length       : 0x00000038 (56)
  Data Type         : 0x00000002 (Message ID)
  Import Source     : 0x00000063 (99)
  Message ID size   : 0x00000028 (40)
  Message ID        : DstTest Message ID 1
```

Data Element (2)

```
  Data Length       : 0x0000015a (346)
  Data Type         : 0x00000004 (XML Metadata)

  <?xml version="1.0" encoding="utf-16" ?>
 <apm schema_version="1" xmlns=http://www.orchestria.com
        xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
        xsi:schemaLocation="http://www.orchestria.com xmleventattributes.xsd">
        <event>
                <file>
                        <host>ta-gf-wxptest</host>
                        [set the File Name variable]Small File.txt </filename>
                        <path>c:\test\data\smarttag</path>
                </file>
        </event>
 </apm>
```

Data Element (3)

```
  Data Length       : 0x00000016 (22)
  Data Type         : 0x00000001 (Data)
  00000000: 4120536d 616c6c20 66696c65 0d0a     : A Small file..  :
```

Footer

```
  Signature         : 0x99999999
External Agent Request ends==========================================
```

External Agent Response Message

This is a sample External Agent Response message to the above External Agent Request Message.

```
Smart Tag Response ================================================
```

Header

```
  Schema         : 0x01001001
  Message Size   : 1132
  Routing ID     : 0
  Message ID     : 1
  Message type   : 0x10002000 (Smart Tag Response)
  Priority       : 0
  Flags1         : 0x01 (Expect Reply)
  Flags2         : 0x00 (No Flags set)
  Flags3         : 0x00 (No Flags set)
  Hr Process     : 0x25631356
  Routing Context: 0x00000000
  Reserved 2     : 0x00000000
  Reserved 3     : 0x00000000
```

Message data

Data Header

```
  Completion ID    : 0x0500e173
  Processing Result: 0x25631356
  Element Count    : 0x00000002
  Reserved 1       : 0x00000000
```

Data Element (1)

```
  Data Length     : 0x00000102 (258)
  Data Type       : 0x00000006 (HRESULT Text)
  Data           : 0x25631356 (The policy engine has successfully completed event
                     processing and is waiting for the caller to commit or abort. )
```

Data Element (2)

```
  Data Length      : 0x0000032c (812)
  Data Type        : 0x00000005 (XML Smart Tags)

<?xml version="1.0" encoding="UTF-16" ?>
<apm schema_version="1" xmlns=http://www.orchestria.com
    xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
    xsi:schemaLocation="http://www.orchestria.com xmleventattributes.xsd">
    <policy>
      <state>
        <smart_tags>
          <smart_tag name="A File Smart Tag">
            <value> Smart Tag generated by Capture File Trigger 1.</value>
          </smart_tag>
        </smart_tags>
      </state>
    </policy>
</apm>
```

Footer

```
  Signature         : 0x99999999
Smart Tag Response ends=========================================
```

External Agent Completion Message

A sample External Agent Completion message sent after the External Agent Response message was received.

```
External Agent Completion=======================================
```

Header

```
  Schema         : 0x01001001
  Message Size   : 128
  Routing ID     : 0
  Message ID     : 1
  Message type   : 0x10003000 (External Agent Completion)
  Priority       : 0
  Hr Process     : 0x00000000
  Flags 1        : 0x00 (No Flags set)
  Flags 2        : 0x00 (No Flags set)
  Flags 3        : 0x00 (No Flags set)
  Routing Context: 0x00000000
  Reserved 2     : 0x00000000
  Reserved 3     : 0x00000000
```

Message data

Data Header

```
  Completion ID   : 0x0500e173
  Completion Flags: 0x0001 (Do Commit)
  Response Flags  : 0x0000
  Element Count   : 0x00000001
  Reserved1       : 0x00000000
```

Data Element (1)

```
  Data Length     : 0x00000044 (68)
  Data Type       : 0x00000002 (Message ID)
  Import Source   : 0x00000063 (99)
  Message ID size : 0x00000034 (52)
  Message ID      : DstTest Compl Message ID 1
```

Footer

```
  Signature        : 0x99999999
External Agent Completion ends====================================
```

External Agent Response Message (Completion)

A sample External Agent Response message to the above request message External Agent Completion message.

```
Smart Tag Response ============================================
```

Header

```
  Schema         : 0x01001001
  Message Size   : 60
  Routing ID     : 0
  Message ID     : 1
  Message Type   : 0x10002000 (Smart Tag Response)
  Priority       : 0
  Hr Process     : 0x00000000
  Flags 1        : 0x00 (No Flags set)
  Flags 2        : 0x00 (No Flags set)
  Flags 3        : 0x00 (No Flags set)
  Routing Context: 0x00000000
  Reserved 2     : 0x00000000
  Reserved 3     : 0x00000000
```

Message data

Data Header

```
  Completion ID    : 0x0600e173
  Processing Result: 0x00000000
  Element Count    : 0x00000000
  Reserved 1       : 0x00000000
```

Footer

```
  Signature        : 0x99999999
Smart Tag Response ends=============================================
```

External Agent Cancel Message

A sample External Agent Cancel message.

```
External Agent Cancel===============================================
```

Header

```
  Schema         : 0x01001001
  Message Size   : 128
  Routing ID     : 0
  Message ID     : 1
  Message type   : 0x10004000 (External Agent Cancel)
  Priority       : 0
  Hr Process     : 0x00000000
  Flags 1        : 0x00 (No Flags set)
  Flags 2        : 0x00 (No Flags set)
  Flags 3        : 0x00 (No Flags set)
  Routing Context: 0x00000000
  Reserved 2     : 0x00000000
  Reserved 3     : 0x00000000
```

Footer

```
  Signature      : 0x99999999
External Agent Completion ends======================================
```