



ArgusConnect Pty Ltd
ABN 63107558387
Suite 4, GreenHill Enterprise Centre
University Drive, Ballarat Victoria 3353
Tel: 0415 645 291
Website: <http://www.argusconnect.com.au/>

Ways of communicating with Argus

December 2006

1	Introduction	3
2	Terms and Abbreviations	3
3	Issues to be considered	3
3.1	How will the data be received?	3
3.2	What format is the source data?	3
3.3	What data is needed for patient matching?	3
3.4	Primary objectives	4
4	Calling Argus to send a document	4
4.1	Technique 1: making direct calls to Argus through its API	4
4.1.1	<i>\\Argus\Server\API\Argus API\ArgusWord</i>	4
4.1.2	<i>\\Argus\Server\API\Argus API\Calling Argus via COM</i>	4
4.1.3	<i>\\Argus\Server\API\Argus API\Calling Argus via DLL</i>	4
4.2	Technique 2: Making MAPI calls	5
4.2.1	<i>Placing identifying fields in the body of the Email</i>	6
4.2.2	<i>Placing labels inside the letter</i>	6
4.3	Technique 3: dropping files	6

1 Introduction

Argus provides a complete, secure Email-based infrastructure for the transmission of clinical data. Third party programs can utilise the Argus infrastructure by:

- making direct calls to Argus through its API
- standard MAPI calls
- dropping files in a directory for Argus.

2 Terms and Abbreviations

Term	Definition
Argus	The suite of programs that provides for the secure mail-exchange between healthcare providers (GPs, pathology labs, psychiatrists etc.)
ArgusMessenger	This is a program that interacts with POP3/SMTP, IMAP and LDAP servers. It implements advanced HL7 handling and encryption via PKI. One instance of ArgusMessenger runs in each practice.
ArgusAgent	This is a program that polls designated directories, looking for files that have been dropped for Argus to detect and send.
ArgusAPI	This is an API module that allows applications to interact with Argus without using the Argus GUI. Windows applications can use a COM object, and non-windows applications can interact directly with a Java class inside Argus itself.
Message	A mail item downloaded from the mail server and stored in an Interbase database. Messages may include attachments.

3 Issues to be considered

3.1 How will the data be received?

In a medical environment the process of sending data must be guided not so much by focussing on the sender, but on the workflow of the recipient. All decisions about how to package data for transmission must consider how the data is to be received. Since the market in Australia is currently dominated by Medical Director (MDW), this dictates that non-results must usually be in PIT format in order that these data can be matched automatically with the patient record for which the data is relevant.

3.2 What format is the source data?

Another complication is that most clinical packages save letters created with their letter writer modules in RTF format, making it difficult to extract data pertaining to the recipient of the message. Also, some packages have trouble displaying RTF data in their letter readers.

3.3 What data is needed for patient matching?

A letter can be uploaded in the same way that pathology results can be uploaded: by being dropped into the upload directory as a PIT file. In order to be matched to a patient, these fields must be defined in the PIT:

- Patient name (*surname, first name*)
- Patient date of birth(*dd/mm/yyyy*)

- Receiving Doctor (*Dr A Doctor*)
- Sending Doctor (not mandatory)

In other words, if a program is to send such data it must know these fields in order to construct a PIT letter that can be uploaded by the receiver.

3.4 Primary objectives

Most applications are already MAPI-enabled, but this function is under-utilised because of a lack of these critical provisions:

- Security of data in transit, using encryption algorithms such as PKI or PGP
- A guarantee of delivery by means of an automatic acknowledgement system
- A cheap way of transmitting data without paying on a per-message basis.

Argus provides a way of addressing all three of these requirements. It uses HeSA PKI for security, the HL7 protocol (ACKs) to ensure delivery and Argus does not charge per message by redirecting all email to a gateway. Transmission is secure, point-to-point and uses standard Email protocols (POP3, IMAP and SMTP).

4 Calling Argus to send a document

Although more than one alternative is explained here, the best approach is to make direct calls to the Argus API; this decision will no doubt be governed by business rather than technical considerations.

4.1 Technique 1: making direct calls to Argus through its API

The Argus API is a Java class inside ArgusMessenger that can be accessed directly, or by calls made to a COM object that was written in Delphi. The API implements 30 functions that allow third-party programs to execute full control over Argus' core functions. This method of controlling Argus is the most secure and efficient, but imposes a coding overhead on the calling program.

A number of fully-functional example programs that use the API are deployed with Argus. Example programs exist in these directories:

4.1.1 `\Argus\Server\API\Argus API\ArgusWord`

This contains a Word template containing VBA code.

4.1.2 `\Argus\Server\API\Argus API\Calling Argus via COM`

This contains a Delphi program that demonstrates how to call the COM object.

4.1.3 `\Argus\Server\API\Argus API\Calling Argus via DLL`

This contains a Delphi program that demonstrates how to call the API, using a standard DLL.

Full documentation for the API can be found in `\Argus\Server\API\Argus API\Documentation`. These are the steps required to send a document via Argus (all code fragments are in Delphi):

1. The user types a letter and clicks a 'Send' button in their application.

2. A variable of variant type must be defined to hold the reference to the Argus API

```
foArgus: variant;
```

3. Retrieve the Argus API COM object

```
foArgus:= CreateOleObject('ArgusAPI_4_2.APIWrapper');
```

4. Initiate the Argus API if a connection to has not been previously initiated

```
foArgus.connectToAPI;
```

5. Log user into Argus

```
if foArgus.isConnected then begin
  showMessage( 'Already connected to database' );
else
  iErrorCode:= foArgus.connectUserToDatabase([IP Address], [Database], [User Login], [Password]);
end;
```

6. Drop a new message into the Argus database.

A variety of functions exist for this step. Some allow the calling program to construct a PIT file, while others allow a text or RTF file to be passed to Argus in such a format that Argus can perform the PIT construction itself. Refer to JArgusAPI.html for a full description of the available functions.

For example, if the calling program creates an RTF and a JPEG file, these can be wrapped up in an HL7 message by means of a call to the Argus API by using:

```
foArgus.addWrappedDocuments
([Sender], [Subject], [Body Text], [recipients], ", ",
[filenames], [OBX segment labels], [encrypt], [sign],
[resends], [resend frequency], -7, "
);
```

Note: There is a caveat inherent in this: the message that is sent is an HL7 message. This allows the ACK process to activate, but it also means that the recipient of the message must use Argus to extract the PIT for upload. Argus can be configured to extract either

- the original RTF
- a PIT file in which the RTF has been embedded
- a PIT file in which the RTF has been embedded as plain text

If the receiver does not use Argus then it has no means of extracting the PIT (or RTF) data from the received HL7 message. In this case the sender cannot wrap the data as HL7; there will be no ACK process, but the message can still be sent encrypted by using:

```
foArgus.addMessage
(
[Sender], [Subject], [Body Text], [Attachments], [Recipients], ", ",
[Encrypt], [Sign],
false, 0, [ConfirmReading], [ConfirmDelivery], false, 0, "
);
```

This function is provided purely as a courtesy and should never be used for clinical data.

7. Send the message

The process of sending is activated only by ArgusMessenger.

4.2 Technique 2: Making MAPI calls

Argus can be installed to function as the default MAPI client on a PC so that Argus will handle all MAPI calls, instead of these being handled by the default mail client (usually Microsoft Outlook). As outlined

above, a critical issue to consider is that the receiver of a letter must be able to upload the letter seamlessly, which requires automatic matching with the patient record, hence PIT data.

This means that when Argus intercepts a call to the MAPI function MapiSendMail(...) it needs information that will allow it to create the appropriate PIT. This can be accomplished in two ways:

4.2.1 Placing identifying fields in the body of the Email

This creates a small coding overhead but offers a powerful way to ensure that data is handled appropriately. The calling program simply needs to insert some identifying fields into the body of the email. For example, the body text might be

Patient Name: ELKERTON,GARY
 Patient DOB: 23/03/1965
 Recipient: Dr Ludwig von Mises
 Sender: Dr Paul Samuelson

With this information in the body of the email, and a reference to the file to be attached (in plain text format) Argus will be able to construct an HL7 message in which a block of PIT is embedded. The file created by the letter writer will be passed in the parameters of the MAPI call.

Once again, the issue of whether the recipient uses Argus is relevant. If not, Argus can simply attach the file created by the letter writer to an ordinary Email, thereby disabling the ACK process. This will be facilitated by the Argus dialog that appears in response to the MAPI call having a checkbox that will allow the user to disable the 'Guarantee of Delivery' option.

4.2.2 Placing labels inside the letter

A way to remove the need for any coding to be done is for the identifying data in the letter to be labelled in such a way as to make it easy for Argus to identify and strip from the letter. This could be accomplished by having users configure a standard letter template in their letter writer that includes labels in the text. Such a template could contain these two lines anywhere in its text:

Patient Name: ELKERTON,GARY	Patient DOB: 23/03/1965
Recipient: Dr Ludwig von Mises	Sender: Dr Paul Samuelson

Argus would simply search through the text of any letter it was asked to send, searching for the identifying labels ('Patient Name:', 'Patient DOB:', 'Recipient:', 'Sender:'). If such labels were found, Argus would have sufficient information to enable it to construct an HL7 message containing an embedded PIT message.

If these labels were not present, Argus would simply treat the letter as it would any attachment. It would be sent without any HL7 wrapping, hence have no ACK mechanism for guaranteeing delivery.

4.3 Technique 3: dropping files

ArgusAgent is a program that polls designated directories for files. For example, as at December 2006, Genie and ZMed have the ability to drop a PIT file whenever a letter must be sent via Argus. The process is as follows:

- User types a letter in their letter writer
- A PIT or HL7 file is dropped into a designated directory
- ArgusAgent detects the file
- ArgusAgent opens the file and extracts details about who to send it to (using either a provider number or email address embedded inside the data)
- ArgusAgent drops the file into Argus' *Outbox*, where it waits to be sent by ArgusMessenger.

Another reason to use ArgusAgent is when the data being dropped is in a proprietary format (neither PIT nor HL7). In such a case, ArgusAgent can be customised with a special 'Event' that will convert the data into an acceptable format before it gets passed to ArgusMessenger for sending.

-- fin ---