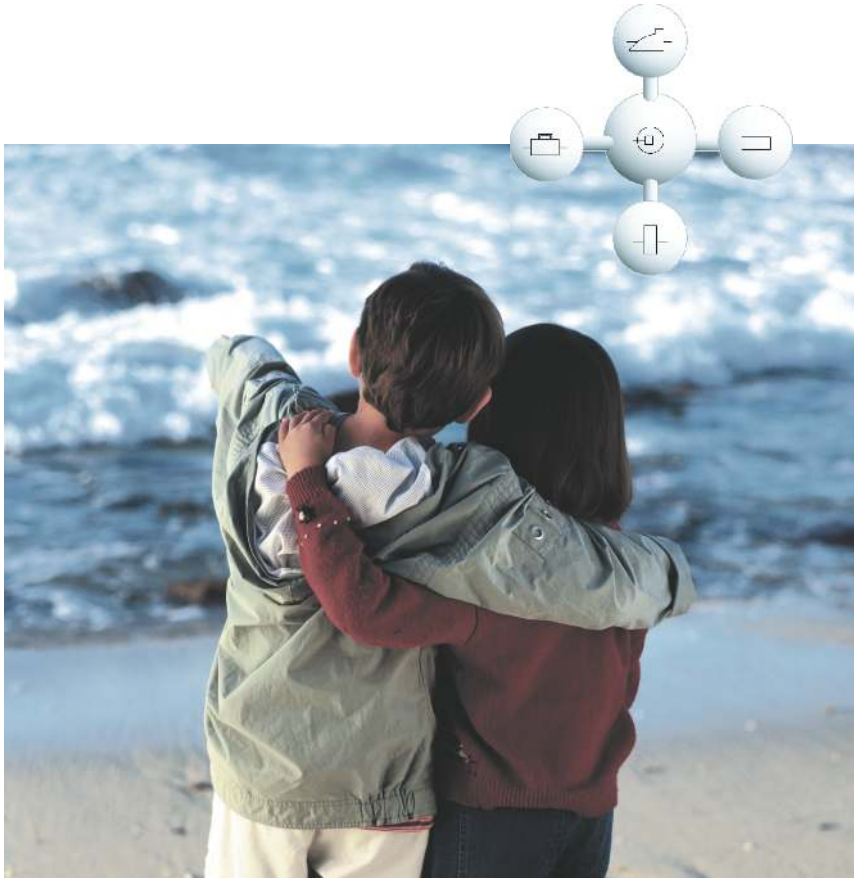
POINT OF CARE
TESTING

DataCarePOC

HIS/LIS Connection Engine
Support Guide



Roche

Roche Diagnostics GmbH
D-68298 Mannheim / Germany
www.roche.com

Rev. 2.0, September 2003

Erstausgabe: Jänner 2002

Rev. 2.0, September 2003

First edition: January 2002

Table of Contents

Introduction

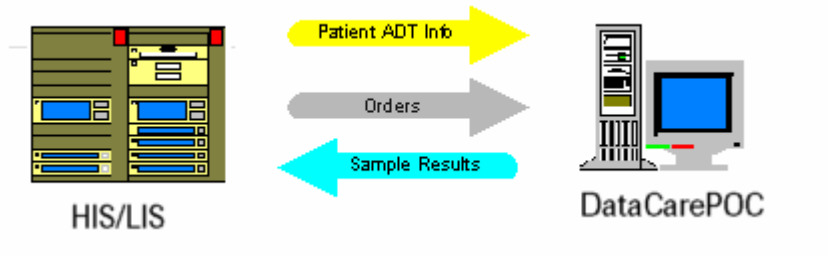
HIS.exe is a program used to interface DataCarePOC with an HIS (Hospital Information System) or an LIS (Laboratory Information System). It facilitates the sharing of information between the DataCarePOC database and an HIS/LIS. It can be used with virtually all HIS and LIS systems on the market today. It also supports different means of physically connectivity and interface protocols.

The described version of HIS.exe is designed to specifically work with the DataCarePOC 32-bit program. However, HIS.exe will need to be configured differently depending on the version of DataCarePOC being used.

The default HIS.exe installation provides a "starting point" for support personnel. Additional customization will likely be required to interface DataCarePOC with an HIS/LIS. Due to the required interaction between the hospital IT personnel, the HIS/LIS vendor, and the Point of Care Coordinator, be aware that some HIS.exe customizations will require extensive coordination and planning.

What Does an Interface Provide?

Interfacing DataCarePOC with an HIS/LIS allows the sharing of information. ADT (Admissions, Discharge, and Transfer) information, orders, sample and QC results can all be shared. This exchange of the information eliminates the need to manually enter the same information into two separate systems. This can save time, reduce work requirements, avoid errors and help keep the information consistent.



Admission, Discharge, and Transfer (ADT) Information

Admissions Admission information entered into the Hospital Information System can be sent to DataCarePOC. DataCarePOC will then create a patient record in its database automatically. This eliminates the need to manually enter a patient's information into DataCarePOC when running a sample.

Discharge The discharge record can be sent to DataCarePOC when a patient is discharged from the hospital. DataCarePOC will then determine if a sample is stored for this patient. If no sample was run, the patient can be automatically purged from the DataCarePOC database. This keeps the database free of unnecessary patient records.

Transfer When a patient is transferred to a different location, this information can be sent to DataCarePOC. DataCarePOC will then automatically update the patient's record in its database. This eliminates the need to manually update each patient's record to reflect his or her current location.

Orders & Results

Orders When a test is ordered within the HIS/LIS, order information can be sent to DataCarePOC. An order is automatically created by DataCarePOC and listed in an Orders Pending List. The user can then select the order from the list and have all the patient and order information needed. The user would not have to search for a patient or manually enter the order into DataCarePOC.

QC Results When a QC result is processed by DataCarePOC all the pertinent sample information can be made immediately available on the HIS/LIS. This allows for tracking of the QC results and validation of patient sample results.

Patient Results When a result is processed by DataCarePOC all the pertinent sample information can be made immediately available on the HIS/LIS. This allows for printing of the results to the proper patient floor locations and immediate charting and billing.

How does HIS.exe work?

The HIS.exe program is used for communication between the DataCarePOC database and an HIS/LIS. One can think of HIS.exe as a translator. It is listening to the HIS/LIS data stream for information relevant to DataCarePOC. It takes in the information (ADT/Orders) and translates it into (name fields) a format used by the DataCarePOC database. It also translates information (QC & patient results) from DataCarePOC into a format that the HIS/LIS can understand. That format is defined by an interface protocol. Standard interface protocols provide a common "language" to allow software systems and instruments from different manufacturers to communicate with each other. HL-7 (TCP/IP) or ASTM (Serial) protocols are two separate industry standards commonly used to interface instruments and data management systems with an HIS/LIS.

HL-7 (Health Level Seven Standard for Electronic Data Exchange in the Health Care Environment) is normally used with data management systems. HL-7 allows for connectivity with an HIS/LIS via a TCP/IP (over a network) connection.

ASTM is often used with instruments. ASTM only allows connection to be made via a serial connection (serial cable). Even though HL-7 and ASTM are defined protocols almost all hospitals implement these protocols differently. HIS.exe is very flexible and was designed with customization in mind. It can easily support variations of both protocols. It can also be customized to handle readable text and new protocols. However, approximately 95% of all interface installs will involve either HL-7 or ASTM or some variation of the two. In the rare case where HIS.exe cannot be used to interface DataCarePOC with an HIS/LIS there are alternatives. A file based method or a scripted keystroke emulation method may be used.

Installation of HIS.exe

Installation Requirements

Determining the requirements for a particular installation is a necessary before installation of the HIS.exe. You may need to consult with the hospital IT personnel and the HIS/LIS vendor. You will need to know the following:

1.	Which HIS/LIS is being used at the hospital.
2.	How the system will be physically connected. (This will usually tell you the protocol you will be using.)
3.	Which protocol will be utilized for the interface.
4.	If the HL-7 (TCP/IP) protocol will used, you will need to know: • The port number that the DataCarePOC Server will use to receive ADTs and Orders from the HIS/LIS. • The IP address of the host HIS/LIS or the information system to which the results will be sent. • The port number used by the host HIS/LIS or information system to which the results will be sent.
5.	If the ASTM protocol will be used, you will need to know: • The connection details, e.g. speed (baud rate) supported. • Which com port the DataCarePOC Server will be using.
6.	If the protocol at the hospital has been implemented in a non-standard way. If so, how it differs.
7.	How ADT, orders, and results are configured on the HIS/LIS.

The DataCarePOC 32-bit server and all its services should be running for the HIS.exe program to function properly.

If the HIS.exe interface is not installed as a service, then the installer must provide a way for the program to start up and run automatically. Two methods of starting the HIS.exe program automatically are:

- To add the HIS.exe program to the startup.bat file
- To include the HIS.exe program in the Start Menu

Installation Procedure

There are two means of running the HIS.exe installation program.

- The HIS.exe installation program may exist in a sub-directory of DataCarePOC application.
- The HIS.exe installation program can be ordered separately.

Please note that the HIS.exe installation program is supplied with the DataCarePOC 32-bit program.

The installation program is preinstalled with DataCarePOC and is located in:

- C:\Program Files→ Roche Diagnostics → DataCarePOC → Setup → HIS-Setup.exe

Or,

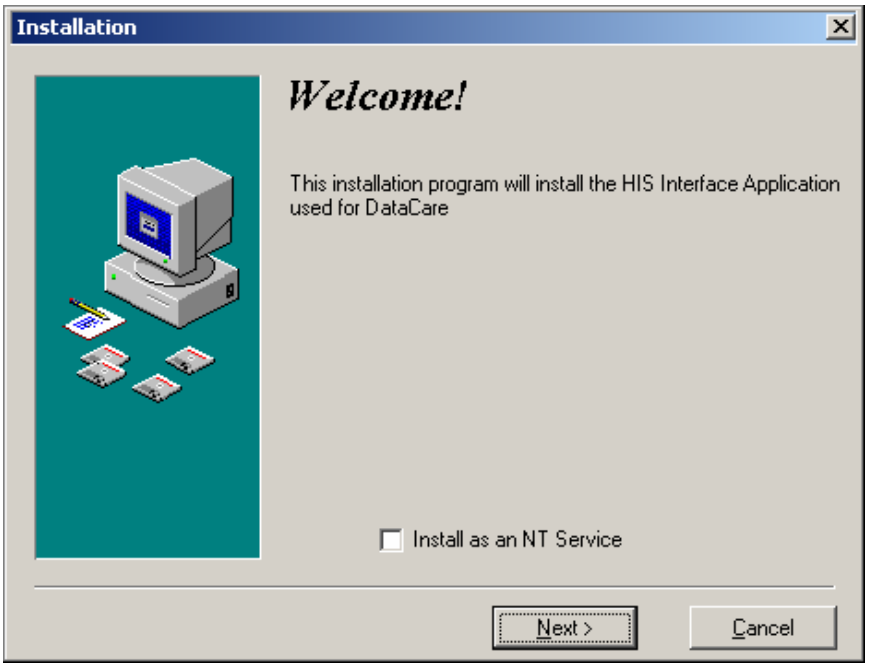
If you have a copy on CD, the file is located at:

- E.g. D:\HIS-Setup.exe

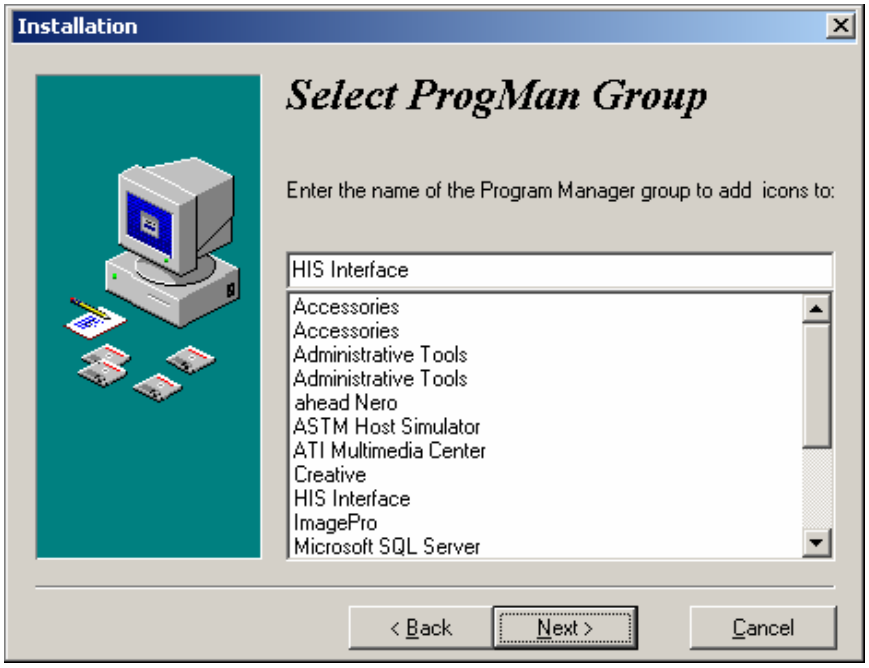
! **DataCarePOC must be installed prior to the installation of HIS.exe.**

To install the HIS.exe application on the DataCarePOC Server, only three steps are required as shown in the following sections.

Step 1 — Selecting the Installation Type

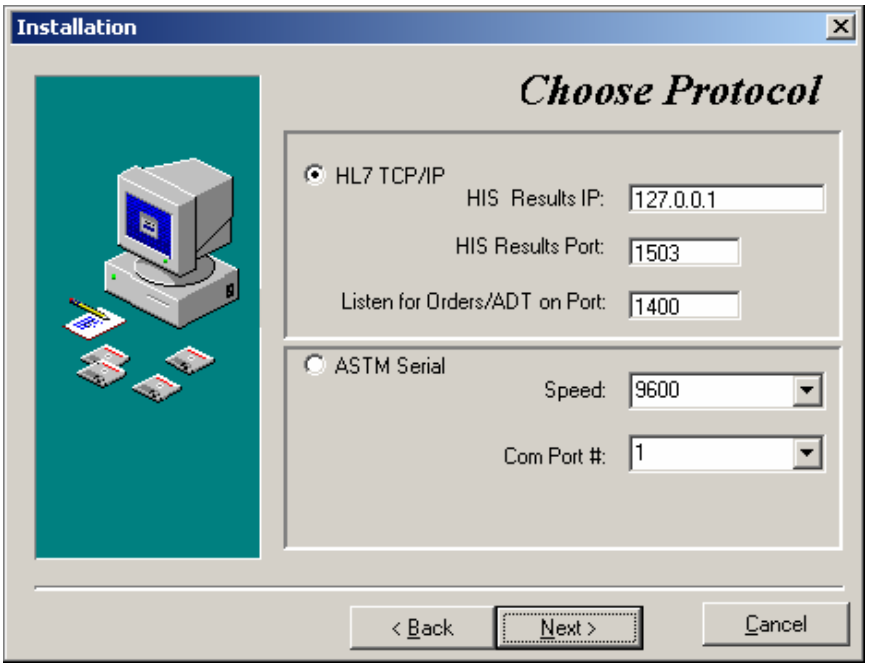
a.	At the DataCarePOC Server PC, log on with administrator privileges and start the setup program by clicking the Windows Start button and then selecting Run. Enter the location of HIS.exe as described above. The following prompt will appear: 
b.	Select "Install as an NT Service" for the HIS.exe program to start automatically when the computer is switched on. Note: To install as a service, you must be logged in with full rights for the Windows operating system. We recommend that you install the program as a service. If you choose not to install HIS.exe as a service, you can add the program to the Start program in the Startup group.
c.	Click the Next button to continue.

Step 2 — Selecting the Program Group

a.	When the **Next** button has been selected on the previous screen, the Select the Program Manager Group displays. Generally, leave as the default "HIS Interface" to maintain the installation standard across DataCare systems. The Select Program Manager Group screen displays as shown: 
b.	Select the **Next** button to continue.

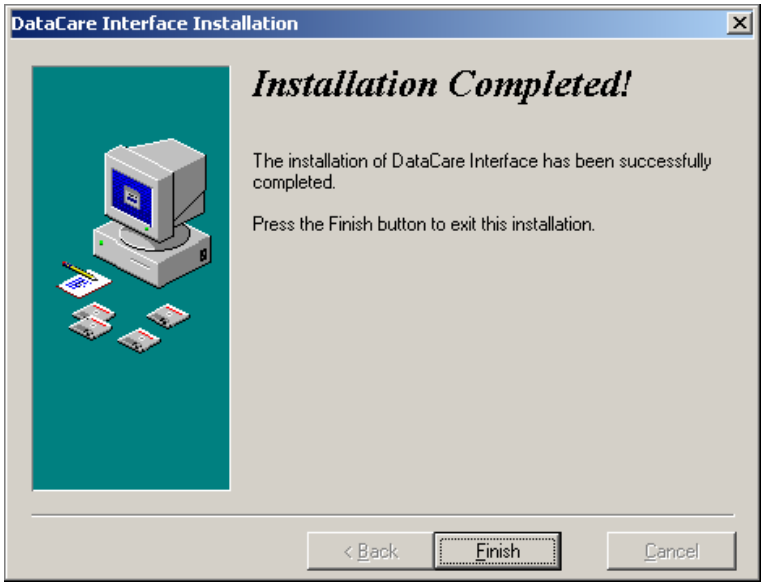
Step 3 — Configuring the Settings

Step three of the installation procedure requires you to configure the HIS.exe settings that will be used in the HIS.ini file.

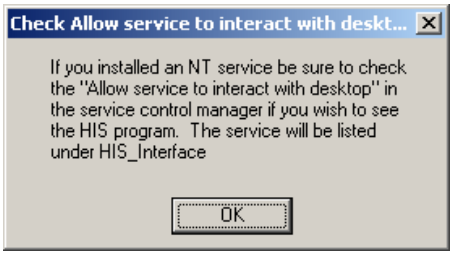
a.	Select either the HL7 TCP/IP protocol or ASTM Serial protocol. 
b.	For HL-7 selections: • Enter the IP address to which DataCare results will be sent. • Enter the host HIS/LIS Results Port information to which DataCare will send results. • Enter the Listener Port information from which DataCare will receive Orders and ADT information (from the HIS/LIS system). These settings details can also be changed later on in the HIS.ini file.
c.	For ASTM Serial Selections: • Enter the Speed of the serial communication (baud rate). • Enter Com Port number of serial connection on the server. These settings details can also be changed later on in the HIS.ini file.
d.	Select the **Next** button to start the Interface installation.

Installation Complete

Once the program finishes customizing the interface files with the protocol and settings selected, the "Installation Complete" screen will display:



If you selected "Install as an NT Service," the following screen will appear as a reminder:



If you did not select "Install as an NT Service," the following screen will appear as a reminder:

Installation of the HIS.exe program is now complete. If the HIS.exe is installed as a service, you need to start the service or reboot. You will now need to configure and customize several files. The files and their settings will be discussed in the following sections.

! In case ASTM was selected, please make sure to copy the settings files from the \ASTM settings files\ directory from the CD to the C:\HIS directory.

Configuring & Customizing the HIS.exe Files

The Files Installed

Once the HIS.EXE files have been installed, the following files will become available in the **C:\HIS** directory.

HIS.EXE	- Interface Engine
His.ini	- General Engine related configurations
Adt.ini	- ADT and Orders definitions
Results.ini	- Patient result side definitions
QC.ini	- QC result side definitions
Tables.ini	- Lookup tables used by both ADT and results
Mscomm32.ocx	- Serial Port component
IADTps.dll	- Proxy stub for ADT/Orders link to HL7x.exe
Iresultsps.dll	- Proxy stub for Results link to HL7x.exe
Profiler.exe	- A GUI program for editing the Results.ini and tables.ini files ***(not recommended to be used with this version of HIS.exe)***
Default.dat	- Upload codes as seen by the HIS
Format.dat	- Format of the data
Lbnames.dat	- Real world names given to the codes
Ranges.dat	- Static ranges for the upload codes
Type.dat	- Type of the value (*i.e.*, Query)
Units.dat	- Static units for the upload codes
Values.dat	- DataCarePOC 32 names for the associated Codes
HisEmula.exe	- Emulates an HIS system (test simulated HIS interface)

The Default Installation

The choices you made in the program setup will be reflected in some of the files that are installed. There are different default installations depending on your choices made for the protocol. You may need to make some minor modifications depending on how the hospital implemented the protocol you are using and how ADT, orders, and results were configured on the HIS/LIS.

Modifying Configuration Files

The configuration files are the initialization (ini) files and can be modified in a simple text editor such as Windows Notepad.

It is important to understand which values or settings can be used in each of the ini files. The files and their settings are discussed in detail below.

HIS.ini

Field	Example	Definition
ProtocolMode	HL-7TCPIP	The mode in which the interface operates. Valid values are: ASTMSerial HL7TCPIP
RemoteIP	127.0.0.1	IP address of the system receiving the results
OutPort	1503	Port number on which the results are being received
InPort	1400	Port number on which HIS.ini listens for ADT/Orders
ADTIn	0	Number of ADT messages received to date
OrdersIn	0	Number of orders received to date
ResultsOut	0	Number of results received to date
ErrorCount	0	Number of errors that have occurred to date
LogOrders	0	Log orders to the his.log file. 0=Do not log orders 1=Log orders
LogResults	0	Log Results to the his.log file. 0=Do not log results 1=Log results
LogEvents	1	Log Events to the his.log file. 0=Do not log Events 1=Log Events
LogCapture	0	Capture and log entire communication
ActiveX	0	Use the ActiveX link ActiveX.ocx 1=DataCare 0=DataCarePOC 32
ResultsCheckTimer	100	How often results should be checked 100=1 sec
UseSerialPort	0	Serial Port Active 0=Serial 1=TCPIP
SerialPort	1	Windows-assigned Com port Number (For ASTM serial)
SerialPortString	N,8,1	Parity, DataBits, StopBits
SerialPortBPS	9600	Speed in Baud Per Second

Tables.ini

The Tables.ini file is used by qc.ini, results.ini and adt.ini. Most values received from and sent to the HIS system can be passed through tables. Tables are useful when the type of information differs from what is expected. One common example of this is the sex of the patient. In the DataCarePOC 32-bit application the patient's sex is defined as M, F, or U. A hospital HIS system might send the entire word for the patient's sex (Female, Male) instead of using M, F, or U. To compensate for this mismatch, a table is created.

```
[TBLSex]
-Default=U
Female=F
Male=M
Other=O
Unknown=U
M=M
O=O
U=U
F=F
```

The Table name is "TBLSex," which will be referenced to by items in adt.ini. Another table may be needed for the reverse translation for the results.ini. Notice the –Default (in line two above). All Tables need a –Default value. If an Item is not found in the table the value for the default is used.

Another useful pre-defined table is Flaglook:

```
[Flaglook]
-Default=?
A=A
B=B
C=C
D=D
```

The Flaglook table is used to translate DataCarePOC 32 flags to HIS/LIS flags. Notice the default settings. Every value translates to itself (A=A, B=B, etc.). It was created as a placeholder. An example from a DataCarePOC 32 system would be.

```
[Flaglook]
-Default=OK
A=Above normal
B=Below normal
C=Above panic
D=Below panic
```

The above table might be useful when customizing a DSP type interface.

ADT.ini

[Transaction Type]

The TransactionType determines if the data is an update, an order, or a discharge. TransactionType is a concatenation of two fields. In the example given below, it is MSH8 + ORC1. The first is used to determine the type and the second is used to determine the action on that type. This is actually used to distinguish a "New Order" from a "Canceled Order."

For DataCarePOC 32-bit, the TransactionType needs to be resolved to a natural number. HIS.exe uses this information to parse out and determine a transaction type using the configuration information defined in the [TransactionType] section of the ADT.ini file. This information about the transaction type is passed to DataCarePOC 32-bit for processing. It may be necessary to change or update this depending on a customers system.

The following Transaction Types are currently defined in DataCarePOC 32-bit:

1 = Order Creation
2 = Order Deletion
3 = Discharge (DataCarePOC will delete the patient from the database if no results exist for that patient.)
4 = Admit
6 = Update

[TransactionType]

This is an example of the TransactionType as defined by the default HL-7 install.

```
TransactionTypeSegment=8
TransactionTypeLine=MSH
TransactionTypeSegmentAddition=1
TransactionTypeLineAddition=ORC
Default=6
ADT^A03=3
ORM=1
ORMNW=1
ORMCA=2
```

In the above example, if an ADT message cannot be identified the default = 6 or Update will be used. The only ADT Message defined is A03 = 3 or discharge. All other ADT messages will be assigned as an update = 6. When configuring ADT messages, all ADT transactions should be defined as a Discharge, Admit, or Update.

Messages received by the HIS.exe over the interface will be parsed. The ADT.ini file contains [ADTxxx] segment where x is a number. Processing starts at [ADT1] and stops when a consecutive number is reached that does not exist.

Example:
[ADT1]
……..
[ADT2]
……
[ADT3]
……..

In the above example [ADT4] does not exist so processing stops after [ADT3].

Example:
[ADT1]
……..
[ADT2]
……
[ADT4]
……..

Notice, in the above example [ADT3] is missing. The numbers must be consecutive for processing to occur. Only [ADT1] and [ADT2] will be processed. [ADT4] will not.

[ADTxx] Sections

ADT sections contain a single Name and one or more of the following: LineID, Segment, Component, and a Lookup. A typical [ADTxxx] section may look like the following:

```
[ADT20]
Name=Locations.Bed
LineID1=PV1
Segment1=3
Component1=2
```

The above configuration pulls the bed number of the patient from the PV1 line, segment 3, and component 2. The Name field signifies that a call to DataCarePOC 32 will associate Locations.Bed with the data found at PV1 segment 3, component 2. In the example below the Patient location is 01.

```
MSH|^~\&|HIS|HOST|||199406100800||ORM|5901|P|2.1|
PID|||115401||JONES^FRED^Q||19631027|M||C
PV1||I|3E^305^01|R|||73123^WELBY^MARCUS
ORC|NW|89003^HOST|||||||199406100745|
OBR||89003^HOST||ARTERIAL
BLOOD|||||||||||||73123^WELBY^MARCUS|
```

The “1” attached to the end of LineID1, Segment1, and Component1 signifies that this is the first try (search) for Locations.Bed. If no data is in the field specified by LineID1, Segment1, Component1 then HIS.exe tries to find the data at LineID2 Segment2 Component2, and so fourth.

```
Example:
[ADT20]
Name=Patients.Bed
LineID1=PV1
Segment1=3
Component1=2
LineID2=PID
Segment2=32
Component2=0
```

The above configuration would read the Bed from PID 32, component 0 only if it did not exist at PV1 3, Component 2.

List of DataCarePOC ADT Fields

Only a limited number of ADT fields can be updated in the DataCarePOC 32-bit system. Below is a list of DataCarePOC 32-bit fields that can be updated from a HIS/LIS system.

DCPOC32 Recognizable Field	Overwrite in Database	With Admit/Discharge /Update Patient	With New/Update/ Cancel Order
Patients.MedicalRecordNumber	No	Create a new patient if it is unknown	Create a new patient if it is unknown REQUIRED INPUT
Patients.PatientId	Yes		
Visits.VisitId	No	Create a new visit if it is unknown	Create a new visit if it is unknown
Visits.AdmitDate	Yes		
Visits.Height	Yes		
Visits.Weight	Yes		
Patients.LastName	Yes		
Patients.FirstName	Yes		
Patients.MiddleName	Yes		
Patients.DateOfBirth	Yes		
Patients.Sex	Yes		
Locations.NurseUnit	No	Must be a known location or blank	Should be a known Location
Locations.Room	No	Must be a known location or blank	Should be a known Location
Locations.Bed	No	Must be a known location or blank	Should be a known Location
Physicians.PhysicianId	No	Create a new physician if it is unknown	Should be a known Physician
Physicians.LastName	No	Create a new physician if it is unknown	Should be a known Physician
Physicians.FirstName	No	Create a new physician if it is unknown	Should be a known Physician
OrderId	No	N/A	Create a new Order if it is unknown
PanelName	No	N/A	Should be a known Panel
TestCode1	No	N/A	Should relate to a known Panel
TestCode2	No	N/A	Should relate to a known Panel

Special Symbols in the Name field

Within the adt.ini files, special symbols are used in the Name field to signify control in the HIS.exe. Currently symbols are used only for graphical user interface (GUI) control.

The only symbol recognized by the HIS.exe program is "!", the exclamation mark. With the "**!**" as the first letter in the name field, HIS.exe takes the data acquired and adds it to its GUI display for that order. Whatever follows "!" is completely insignificant. This "!" symbol is useful because the GUI can be customized to display any field for that transaction. There can be as many "!" as desirable. For each transaction any data picked up by an "!" is shown on a single line separated by a space. DataCarePOC does not recognize the "!" symbol, and the HIS.exe does not pass these fields on to DataCarePOC.

Results.ini

Results side processing is a somewhat different from ADT. On the ADT side, the interface engine (HIS.exe) is basically reading packets and picking out data. The results side deals with defining how a packet looks, how it is sent, what data appears in it, and the format of that data. The Results.ini file contains these sections and can be quite lengthy.

Results components:

[ResultsID]	- Program control switches
[ResultAddx]	- Definition for header (non repeated) type sections
[ResultLineSetupx]	- Definition for the Result Lines
[ResultLinex]	- Actual Result Lines

[ResultsID]

Field	Example	Definition
TCPIP_TransactionTimeOut	50	Number of seconds before an IP packet times out and re-transmits 50=5 seconds
Serial_TransactionTimeOut	300	Number of seconds before a serial bound packet times out 300=30 seconds.
Version	2.2	Version number to store in the Dynamic Version variable
ProcessingID	P	ProcessingID number to store in the Dynamic ProcessingID variable
SerialENQ	5	Char number to use for a Packet start (Serial)
SerialACK	6	Char number to use for a line ack (Serial)
SerialNAK=	21	Char number to use for a line Nack (Serial)
ResultLineID	OBX	ID of the repeating segment. Repeating segments are usually one of the following: OBX (HL-7) DSP (HL-7) R (ASTM)
COUNTER	0	Number to store as a starting point for the Counter Dynamic Variable. The Counter Variable gets incremented after every result Line.
UseStaticRanges	0	0=Use Dynamic Ranges (Those stored with the result) 1=Use Static Ranges
UseStaticUnits	1	Denotes whether to use fixed or Dynamic Units. DataCare always uses Fixed Units.
FieldSeparator	|^~\&	What field separators to use
PacketStart	11	Char number
LineStart		Char number Used for an ASTM type Configuration
LineEnd1	13	
LineEnd2		Char number Used for an ASTM type Configuration
MessageEnd		Char number Used for an ASTM type Configuration
UseCheckSum	0	1=Don't use check sum type packets 0=Use check sum type packets
PacketEnd1	28	Char number used to signify the end of the first part of a line.
PacketEnd2	13	Char number used to signify the last part of a line.

Result Lines

For the purpose of this program, result lines are those that are of the same type and that repeat for different values. This constructs a name value pair. In HL-7, this is the OBX line. Each OBX line contains a name, a value, and other information specifically related to that value. An example of an HL-7 TCPIP result line is the following:

```
OBX|0|ST|GLUCOSE|1|120|mg/dl|30-110|A|||F|20001201085500
```

This line's format is actually defined through the use of [ResultLineSetupxxx] sections. It is completely customizable. Fields can be added, dropped, or moved.

Dynamic Variables

Dynamic variables play an important role in the use of result line setups. The following is a list of dynamic variables available:

Dynamic Variable	Definition
COUNTER	Incremented after every result line
UPLOAD_CODE	Is set to the UploadCode of the current line
VALUE	Is set to the value for which the current line results
RESULTLINE_FORMAT	Is set to the Format of the current line
UNIT	Is set to the Unit of the current line
RANGE	Is set to the Range of the current line
FLAG	Is set to the Flag of the current line
ANALYZEDATETIME	Is set to the AnalyzeDateTime of the Current line
NOW-DATETIME	The current date and time in YYYYMMDDHHNNSS format
MESSAGEID	Message Control ID determined by a query to HL7x.exe
HL7VERSION	Determined by [ResultsID] Version=

[ResultAddx]

This section defines lines that are not repeated in the interface (*i.e.,* MSH, PID, ORC, and OBR). The lines are automatically added when a request for a line ID does not match any that has already been called upon. It is critical that the first instance of each line match the order of the desired output. In HL-7, the MSH line comes before the PID line. The first instance of MSH must exist before the first instance of PID to maintain the correct order.

Note: Any LineID can be used. LineID's do not have to match any protocol, so if a nonstandard one is desired it can be easily added. (*i.e.,* LineID=PVA)

A [ResultAddxxx] section contains a single LineID-segment-component and can contain multiple Type, Value, Lookup, and Format. If the first Type-Value (Type1, Value1, Format1) combination does not yield a result it moves onto the next one. (Type2, Value2, Format2).

Example:

```
[ResultAdd6]
LineID=MSH
Segment=6
Type1=DYNAMIC
Value1=NOW-DATETIME

[ResultAdd15]
LineID=PID
Segment=5
Component=1
Type1=QUERY
Value1=Patients.FirstName

[ResultAdd18]
LineID=PID
Segment=8
Component=0
Type1=QUERY
Value1=Patients.Sex
Lookup1=TBLSex
```

[ResultLineSetupxxx]

ResultLineSetup defines the layout of a result line. The result line is defined by [ResultLineSetupxxx] sections. A [ResultLineSetupxxx] section contains a single segment-component and can contain multiple Type, Value, Lookup and Format. If the first Type-Value (Type1, Value1, Format1) combination does not yield a result it moves onto the next one (Type2, Value2, Format2).

Example:

```
[ResultLineSetup5]
Segment=5
Type1=DYNAMIC
Value1=VALUE
Format1=RESULTLINE_FORMAT
```

Segment:

The segment number to put the data defined through Type-Value-Format.

Type:

Valid types are DYNAMIC and STATIC. If a type is set to DYNAMIC, the Value must be one from the defined list of dynamic variables.

Value:

If a type is set to Dynamic, a dynamic variable must appear in this field. If the Type is set to STATIC, whatever is placed in the Value field appears in the defined segment.

Format:

Whatever result comes from the Type-Value combination, it is passed through a standard C-Style formatting routine.

%s	Format as a variable length string
%0.1f	Floating point value with one value to the right of the decimal place.

[ResultLinexxx]

[ResultLinexxx] defines the values that are placed in the result line. There is one [ResultLinexxx] for every result line record (*i..e.,* OBX line). This should not be confused with [ResultLineSetupxxx] that deals with the layout of the Result Line.

Each line contains a single UploadCode, Static_Unit, Static_Range but may contain many Query, Format, Lookup, and Type. If (Query1, Format1, Lookup1, Type1) does not yield a value, then processing moves on to Query2, Format2, Lookup2, Type2.

UploadCode, Static_Unit, and Static_Range get moved to Dynamic Variables.

Static_Range is overwritten at this point with the value queried from DataCare if the switch UseStaticRanges=0.

As in the [ResultLineSetupxxx] setup, Typex= can be Static, Dynamic or Query. Queryx replaces Valuex to distinguish differences. If set to Query with DCPOC 32-bit, Queryx is the Name DataCarePOC 32-bit defines for the data (see “List of DataCarePOC ADT Fields”).

Example:

```
[ResultLine63]
LINEID=----------ResultLine63----------
UploadCode=PHM
Static_Unit=
Static_Range=
Query1=PH
Format1=%0.3f
Lookup1=
Type1=QUERY
```

Note: For DataCarePOC, query names must match from names defined in database. In addition, format must be defined as expected in the HIS/LIS system. For example PH and PHM above will report with 3 places to right of the decimal.

List of DataCarePOC 32-bit Patient Result Query NAMES.

This list is by no means complete list of all defined NAMES to be used for DataCarePOC 32-bit. New NAMES will be introduced in new versions of the DataCarePOC 32-bit application. A default list is located in the directory where the HIS adapter is stored (c:\his) in a file called default.dat. The values.dat is a file used by the Profiler program to aid in interface customization.

DateCarePOC32 (values.dat)	Format (Format.dat)	DateCarePOC32 (values.dat)	Format (Format.dat)
Patients.MedicalRecordNumber	%s	BO2	%s
Patients.PatientId	%s	ctO2	%s
Patients.LastName	%s	PAO2	%s
Patients.FirstName	%s	PAO2t	%s
Patients.MiddleName	%s	AaDO2t	%s
Patients.DateOfBirth	%s	a/AO2t	%s
Patients.Sex	%s	a/AO2	%s
Patients.Aux1	%s	RI	%s
Patients.Aux2	%s	RIt	%s
Visits.VisitId	%s	C	%s
Visits.AdmitDate	%s	BUN(c)	%s
Visits.Height	%s	Qt(s)	%s
Visits.Weight	%s	P/F Index(s)	%s
LocationName	%s	avDO2	%s
Locations.NurseUnit	%s	OER	%s
Locations.Room	%s	Qs/Qt	%s
Locations.Bed	%s	Qt	%s
Physicians.PhysicianId	%s	P/F Index	%s
Physicians.LastName	%s	MCHC	%s
Physicians.FirstName	%s	Osm	%s
Physicians.MiddleName	%s	nCa++	%s
OrderId	%s	st.HCO3-	%s
AnalyzerName	%s	cHt	%s
AnalyzerSerial	%s	Accepted	%s
OrderingPhysicianId	%s	Date Analyzer	%s
OrderingPhysicianFirstName	%s	Date Reported	%s
OrderingPhysicianLastName	%s	Insurance No	%s
OrderingPhysicianMiddleName	%s	Time Analyzer	%s
PanelName	%s	Time Reported	%s
TestCode1	%s	MiddleInitial	%s
TestCode2	%s	Custom 3	%s
OrderDateTime	%s	Bypass	%s
SpecimenType	%s	Pplat	%s
SpecimenSource	%s	PS	%s
UserId	%s	IE	%s
CollectUserId	%s	BiLevel	%s
CollectTime	%s	Sex	%s
CollectDate	%s	Blood Type	%s
CollectDateTime	%s	Technician	%s
InstrumentTestDateTime	%s	Time of Collection	%s
InstrumentTestDate	%s	Date of Collection	%s
InstrumentTestTime	%s	Custom 1	%s
LotNUmber	%s	Custom 2	%s
ReviewStatus	%s	Comments	%s
OrderStatus	%s	Department	%s
Comment1	%s	TVOL	%s
Comment2	%s	MVOL	%s
Comment3	%s	LPM	%s
Comment4	%s	ITIME	%s
COALESCE_pH(t)_pH	%s	ETIME	%s

COALESCE_PCO2(t)_PCO2	%s
COALESCE_PO2(t)_PO2	%s
COALESCE_SO2(c)_SO2	%s
pH	%s
PCO2	%s
PO2	%s
Na+	%s
K+	%s
Cl-	%s
Ca++	%s
tHb	%s
O2Hb	%s
COHb	%s
MetHb	%s
HHb	%s
SulfHb	%s
Lac	%s
SO2	%s
Hct	%s
Myog	%s
Trop T	%s
DDimer	%s
Urea	%s
IQC	%s
BUN	%s
H+	%s
Bili	%s
Glucose	%s
ACT	%s
APTT-Ratio	%s
APTT-Secs	%s
PT-Secs	%s
PT-Ratio	%s
PT-INR	%s
PT-%Quick	%s
PTn-Secs	%s
PTn-Ratio	%s
PTn-INR	%s
PTn-%Quick	%s
pH(t)	%s
PCO2(t)	%s
PO2(t)	%s
BE	%s
BEact	%s
BEecf	%s
BB	%s
SO2(c)	%s
SRATE	%s
ARATE	%s
a/f	%s
R	%s
P50	%s
24h Urine	%s
MAP	%s
Hb Factor	%s
Billing code	%s
Marital status	%s
VE	%s
VMode	%s
VT	%s
O2Mode	%s
AccessionNo	%s
Attending Physician	%s
FirstName	%s
LastName	%s
Location	%s
PatientID	%s
Sample Type	%s
MCHC%	%s
Puncture Site	%s
RQ	%s
PEEP	%s
PIP	%s
CPAP	%s
DOB	%s
f	%s
FiO2	%s
FR	%s
AllenTest	%s
Baro	%s
Temp	%s
cTemp	%s
ctHb(I)	%s
tCO2	%s
Hct(c)	%s
cH+	%s
AG	%s
AaDO2	%s
P50(c)	%s
HbI	%s
HCO3-	%s
st.pH	%s
ctCO2(B)	%s
ctCO2(P)	%s
FO2Hb	%s

QC.ini

QC results side processing is basically similar to patient results side processing, therefore basically the same structure applies to it. Settings not specifically done in the QC.ini are taken from the Results.ini.

[ResultsID]

Field	Example	Definition
NegativeMessageId	0	Internally HIS.exe is using negative numbers to identify QC messages. When set to 0, the non-negative value is used.
MessageIdFormat	QC%s	To be able to differentiate between patient and QC results, QC is added to the Message ID number.

List of DataCarePOC 32-bit QC Result Query NAMES.

This list is by no means complete list of all defined NAMES to be used for DataCarePOC 32-bit. New NAMES will be introduced in new versions of the DataCarePOC 32-bit application.

DateCarePOC32	Format	DateCarePOC32	Format
LotNumber	%s	tHb	%s
LotControlName	%s	O2Hb	%s
LotControlType	%s	COHb	%s
LotExpirationDate	%s	MetHb	%s
LotNumber2	%s	HHb	%s
LotControlName2	%s	SulfHb	%s
LotControlType2	%s	Lac	%s
LotExpirationDate2	%s	SO2	%s
QCLevel	%s	Hct	%s
Comment1	%s	Myog	%s
Comment2	%s	Trop T	%s
Comment3	%s	DDimer	%s
Comment4	%s	IQC	%s
CommentX	%s	Urea	%s
pH	%s	BUN	%s
PCO2	%s	Bili	%s
PO2	%s	Glucose	%s
Na+	%s	ACT	%s
K+	%s	APTT-Secs	%s
Cl-	%s	PT-Secs	%s
Ca++	%s	PTn-Secs	%s

Other Files Installed

IADTps.dll and iResultsps.dll

The IADTps.dll and iResults.dll files are used with the HL7x.exe program to interface the DataCarePOC 16-bit program with an HIS/LIS.

activeX.ocx

ActiveX components (activeX.ocx) is the HIS.exe interface link to DataCarePOC 16-bit using these dll files as the proxy stubs for connecting to the correct systems for accessing the data. DataCarePOC 32-bit uses straight COM and does not use the activeX.ocx. (ActiveX.ocx may be needed to run some of the test programs.)

HisEmula.exe and ActLike3rdPart y.exe

Generally, development and support staff only use these two programs. They are included for troubleshooting purposes, only.

HisEmula.exe emulates an HL-7 (TCP/IP) HIS system. When used with the HIS.exe program, results sent from the HIS.exe to the HisEmula.exe will always be acknowledged, similar to talking to a live system.

ActLike3rdParty.exe is a program that emulates the DataCarePOC 32-bit component on the system. ActLike3rdParty requires configuration of the HIS.ini file in order to run. In the HIS.ini file the following settings must be changed: ActiveX=1. The ActLike3rdParty will continuously send results to the HIS.exe program, as long as it is running. The data is clinically invalid but useful for testing HIS connections. If an ADT or order is sent to HIS.exe, HIS.exe will parse it and pass it on to ActLike3rdParty where the Name Value pairs that DataCarePOC 32-bit would normally see will be displayed.

APPENDIX A – ASTM Interface Specifications

Description This document describes the ASTM protocol as used by DataCarePOC. Regardless of the system being used, the following protocol will be followed.

Cabling Use standard serial cable to connect the DataCare computer to the HOST system. The connection at the DataCare computer will be to the PC's Com port using a standard 9-Pin RS-232 female connector. Only three wires are necessary to effect this connection. The pin configuration on the computer is as follows.

- Pin 2 – Receive
- Pin 3 – Transmit
- Pin 5 – Ground

Serial Interface Specifications

The HOST and DataCarePOC computers will communicate using an RS-232 interface. The low level interface will follow the ASTM E 1394-91 and ASTM E 1381-91 standards as closely as possible. This interface uses a checksum that will be necessary with the RS-232 communication.

The packet of information is formatted:

<STX>{DATA}<ETX>{CheckSum}<CR><LF>

Where:

STX	ASCII character 0x01
{DATA}	The ASTM message with ending CR LF
ETX	ASCII character 0x03
{CheckSum}	Two character checksum
CR	Carriage Return, ASCII character 0x0D
LF	Line Feed, ASCII character 0x0A

The checksum is the sum of all characters starting with the first character after the <STX> through and including the <ETX> character. The resulting sum is modulo 256 and converted to two ASCII characters in the range of "00" to "FF" inclusively.

Record Types

Header Record (H)

Minimal implementation example:

H|\^&<cr>

Extended example:

H|\^&||DATACARE^3.00^12345|||||HOST||P|1|19941101120130<cr>

1. Record Type ID.
   H — Header record
2. Delimiter Definition.
   | — Field delimiter
   \ — Repeat delimiter
   ^ — Component delimiter
   & — Escape delimiter
3. Message Control ID.
   Access Password.
4. Sender Name or ID.
   DATACARE — Instrument Name
   ^
   3.00 — Version Number
   ^
   12345 — Serial Number
5. Sender Street Address.
6. Reserved Field.
7. Sender Telephone #.
8. Characteristic of Sender such as protocol.
9. Receiver ID.
   HOST — Receiver ID
10. Comment or special instructions.
11. Processing ID.
12. ASTM Version #.
    Version level of the ASTM specification
13. Date and time of message.
    yymmddhhmmss — Date and time message was sent.

Patient Information Record (P)

Example Patient Record:

P|1||12345||SMITH^JOHN||19631027|M|||||12345^JONES^WILLIAM<cr>

1. Record Type ID.
   P — Patient Information Record
2. Sequence Number.
   1 — First P record in message
3. Practice Assigned ID number.
4. Laboratory Assigned ID number
   12345 — Lab's patient id number
5. Patient ID
6. Patient Name
   SMITH — Last Name
   ^
   JOHN — First Name
7. Mothers Maiden Name
8. Birthdate19631027 — Birthdate in YYYYMMDD format.
9. Patient Sex
   M — Patient Sex
10. Patient Race
11. Patient Address
12. Reserved Field
13. Patient Telephone
14. Attending Physician
    Physician Number
    ^
    JONES — Physician Last Name
    ^
    WILLIAM — Physician First Name

Order Information Record (O)

Example Order Record:

O|1|H127||^^^ABG|||19941101090000<cr>

1. Record Type ID.
   O — Order Information Record
2. Sequence Number.
   1 — First O record in message
3. Specimen ID
   H127
4. Sequence Number
5. Universal Test ID
   ^^^ABG — Test ordered
6. Priority
7. Requested/Ordered Date and Time
8. Specimen Collection Date and Time
   19941101090000 — Date and Time specimen collected

Result Information Record (R)

Example Result Record:

R|1|^ ^ ^PH|7.394|||||F||12123<cr>

1. Record Type ID.
   R — Result Information Record
2. Sequence Number.
   1 — First R record in message
3. Specimen ID
   ^ ^ ^PH — Universal test id
4. Data Measurement or value
5. Units
6. Reference Ranges
7. Result Abnormal Flags
8. Nature of Abnormality testing
9. Result Status
   F — Final Status
10. Date Change Instrument Normative Units
11. Operator Identification
    12123 — Operator ID

Message Terminator Record (L)

Example Message Terminator Record:

L|1|N<cr>

1. Record Type ID.
   L — Message Terminator Record ID
2. Sequence Number.
   1 — First and Last L record in message
3. Termination Code
   N — N = Normal Termination.

Message Layout

Results Sent To Host

An ASTM packet is sent to the receiving computer via several defined message types. Each record starts with the message type character followed by all the parameters separated by the "|" character. The following example shows a simple blood gas result:

```
H|\^&|||INSTNAME^1^|||||CLIMS-XYZ||P|1|19910123130140P|1
O|1|T5678||||||||||||BLOOD^ARTERIAL
R|1|^ ^ ^PH|7.443|||||F
R|2|^ ^ ^PCO2|29.7|||||F
R|3|^ ^ ^PO2|113.1|||||F
L|1|N
```

Orders Sent From Host

The following example shows an order that is sent from the HOST system to DataCare. DataCare uses the information in order to update it's patient information database.

```
H|\^&|||HOST|||||DATACARE||P|1
P|1|12345|H127||SMITH^JOHN||19701003|M|||||12345^JONES^WILLIAM
O|1|H127||^^^ABG|||19941101090000
L|1|N
```

DataCare uses Patient ID, Name, Date of Birth, Sex, Attending Physician, and Order # from the order message to fill out Patient and Order's tables.

Transfer Protocol

The messages are sent to the receiving system via an ACK/NAK protocol. The owner of the serial line is negotiated by using ENQ/ACK/NAK characters. The system that has a message to send will send an ENQ character to the receiving system. If the receiving system is ready, it will send an ACK character. The sending system will proceed to send each line of the message with a corresponding checksum. The receiving system will then respond with an ACK if the line was processed correctly.

<EOT>	ASCII Decimal 4
<ENQ>	ASCII Decimal 5
<ACK>	ASCII Decimal 6
<ETX>	ASCII Decimal 3
<CR>	ASCII Decimal 13
<LF>	ASCII Decimal 10
<NAK>	ASCII Decimal 21

The following is an example transmission to the HOST system and its responses.

```
<ENQ>
<ACK>
<STX>1H|\^&|||DATACARE^3.00^12345|||||HOST||P|1|19941101120000<CR><ETX>94<
CR><LF>
<ACK>
<STX>2P|1<CR><ETX>33<CR><LF>
<ACK>
<STX>3O|1|T5678||||||||||||BLOOD^ARTERIAL<CR><ETX>20<CR><LF>
<ACK>
<STX>4R|1|^^^PH|7.323|||A||1234<CR><ETX>BA<CR><LF>
<ACK>
<STX>5R|1|^^^PCO2|28.3|||A||1234<CR><ETX>FC<CR><LF>
<ACK>
<STX>6R|1|^^^PO2|92.3|||A||1234<CR><ETX>34<CR><LF>
<ACK>
<STX>7L|1|N<CR><ETX>94<CR><LF>
<ACK>
<EOT>
```

The character preceding each line is a line counter. It is a number from '0' to '7'. The first line sent starts with a '1'.

APPENDIX B – HL7 Interface Specifications

Description This document describes the HL7 interface specifications for the DataCare system. The messages described here are to be considered a starting point, not the final draft, for creating a specification, because the user may define custom features that have to be translated into HL7 messages.

TCP/IP Low Level Interface

The HOST (HIS) and DataCare computers communicate using two TCP/IP sockets. The Order Message socket is initiated by the HOST computer. The DataCare computer opens this socket in a Listen (Server) mode. The Results Message socket is opened in the Connect (Client) mode when the first result becomes available. The HOST computer must have the results socket opened in the Listen (Server) mode. If the DataCare computer cannot connect, it will try to reopen the socket periodically.

The packet of information is formatted:
VT {HL7 Message} FS CR
Where:

VT	ASCII character 0x0B
{HL7 Message}	The actual HL7 message with ending CR
FS	ASCII character 0x1C
CR	Carriage Return, ASCII character 0x0D

The packet information is described further in the HL7 v2.1 interface specifications section B.4 Minimal Lower Layer Protocol. There is no checksum or other method to determine the packet integrity other than that provided by the LAN and TCP/IP software.

Message Acknowledgment

This message is sent to the transmitting computer from the receiving computer upon receipt of a message. The receiving computer can accept or reject the sample.

ACK Message Format

MSH	Message Header
MSA	Message Acknowledgment

MSH - MSH|^~\&|HIS|HOST|||199406101234||ACK|MSG1234|P|2.1|

Where:

HIS	Sending application
HOST	Sending facility
199401101234	YYYYMMDDHHMM Date/Time of message
ACK	Message type
MSG1234	Message control ID
P	Processing ID (P=Production, D=Development)
2.1	HL7 Version ID

MSA - MSA|AA|DC1234|NOERROR|

Where:

AA	Acknowledgment code.
AA	Application Accept
AE	Application Error
AR	Application Reject
DC1234	Message ID of acknowledged message
NOERROR	Error message text.

Results Interface

When a sample is saved in the DataCare system, the Results Interface (RI) will reformat the data from the common HIS format output from DataCare into an HL7 message and transmit the message to the Host system. Results transmitted are patient results and demographics.

ORU Message Format

MSH	Message Header
PID	Patient Identification
ORC	Common Order
OBR	Observation Request
OBX	Results Segment

MSH - MSH|^~\&|DC|RESP|||199406101234||ORU^R01|MSG1234|P|2.1|

Where:

DC	Sending application
RESP	Sending facility
199401101234	YYYYMMDDHHMM Date/Time of message
ORU^R01	Message type
MSG1234	Message control ID
P	Processing ID (P=Production, D=Development)
2.1	HL7 Version ID

PID - PID|||123456789012||PUBLIC^JOHN^Q|

Where:

123456789012	Patient ID
PUBLIC^JOHN	Patient name LAST^FIRST^MIDDLEINITIAL

ORC - ORC|RE|123456^HIS||||||199406100600|

Where:

RE	Order control (RE=Results)
123456^HIS	Placer order #
199406100600	Date/Time of transaction

OBR -

OBR|1|123456^HIS||ART BLOOD|||19951108132900|19951108132900||JM||||19951108132900|R-Radial^^^|||||||19951108132900||||||||||SF|SF

Where:

1Set ID always 1	
123456^HIS	Placer order #
ART BLOOD	Test Code From the Order Sent to DataCare on the Orders Interface
19951108132900	Observation Date/Time AnalyzeDate Start
19951108132900	Observation Date/Time AnalyzeDate End Same as Start
JM	Collector Identifier --Analyzed OpID
19951108132900	Specimen Received Date/Time AnalyzeDate
R-Radial	Specimen Source
19951108132900	Results Rpt Same as AnalyzeDate
SF	OBR-33 Assistant Result Interpreter --Analyzed OpID
SF	OBR-34 Result Interpreter --Analyzed OpID

OBX - OBX|26|ST|PCO2C|1|61.8|mmHg|35.1-45.1|A

Where:

26	OBX Counter
ST	Field type (ST=String) Always ST
PCO2C	Item code (upload Code) (Observation Identifier)
1	Observation Sub-ID Always
61.8	1 Results for the item (Observation Results)mmHg Units
35.1-45.1	Normal Ranges (Reference Range)
	A Abnormal Flags
	Blank Normal Range
	A Critically High
	B Critically Low
	C High
	D Low

ADT Interface

The ADT (Admit, Transfer, Discharge) Interface is responsible for updating Patient information in the DataCare system. Patients discharged from the system are deleted if they have no sample data. The interface follows the HL7 2.1 specification and can receive all the ADT message types defined. This section describes the message types that are processed; all other types are ignored. Additional fields not described in the ADT messages are ignored and will not cause a problem if sent to the DataCare interface.

ADT Events

A01	Admit a Patient
A02	Transfer
A03	Discharge (Deletes from DataCare only if patient has no data)
A04	Register a Patient
A05	Pre-Admit a Patient
A08	Update Patient Information
A11	Cancel Admit
A13	Cancel Discharge
A15	Pending Transfer
A16	Pending Discharge
A18	Merge Patient Information

ADT Messages Required

MSH	Message Header
EVN	Event Type
PID	Patient Identification
PV1	Patient Visit

MSH - MSH|^~\{|ADT||ALL||199511161214||ADT^A01|1234|P|2.1

Where:

ADT	Sending application
ALL	Receiving application
199511161214	YYYYMMDDHHMM Date/Time of message
ADT^A01	Message type
1234	Message control ID
P	Processing ID (P=Production, D=Development)
2.1	HL7 Version ID

EVN - EVN|A01|199511161214

Where:

A01	Event Type Code
199511161214	YYYYMMDDHHMM Date/Time of event

PID - PID|||03047289||NELSON^JODY^|19480513|F|||||||||44447209||

Where:

03047289	Patient ID - Medical Record Number
NELSON^JODY^	Patient Name
19480513	Date of Birth - YYYYMMDD
F	Female
44447209	Patient ID - Account Number

PV1 - PV1||O|ICU^1^2||||01471^JONES

Where:

O	Patient Class - I = Inpatient, O = Outpatient, P = Preadmit
ICU^1^1	Patient Location - NurseUnit^Room^Bed
01471^JONES	Attending Doctor - Doctor#^Doctor Name

Order Interface

The Order Interface of DataCare is responsible for receiving orders. DataCare can use the patient information contained in the order message to fill the Patients table. The Order Interface will also supply information, the order/specimen/accession number needed to be passed back in the Results Interface. The HL7 layout of the message is as follows:

ORM Message Format

MSH	Message Header
PID	Patient Identification
PV1	Patient Visit
ORC	Common Order
OBR	Observation Request

MSH - MSH|^~\&|HIS|HOST|||199406101234||ORM|MSG1234|P|2.1|

Where:

HIS	Sending application
HOST	Sending facility
199401101234	YYYYMMDDHHMM Date/Time of message
ORM	Message type
MSG1234	Message control ID
P	Processing ID (P=Production, D=Development)
2.1	HL7 Version ID

PID - PID|||123456789012||PUBLIC^JOHN^Q||19631027|M||C

Where:

123456789012	Patient ID
PUBLIC^JOHN	Patient name LAST^FIRST^MIDDLEINITIAL
19631027	Date of birth
M	Sex
C	Race

PV1 - PV1||I|3E^305^01|R|||73123^WELBY^MARCUS

Where:

I	Patient status (I=Inpatient)
3E^305^01	Patient location NURSE_STN^ROOM^BED
R	Admission type (R=Routine)
73123^WELBY^	Attending physician DOC_NUM^LAST^FIRST

ORC - ORC|NW|123456^HIS||||||199406100600|

Where:

NW	Order control (NW=New order; CA=Cancel order)
123456^HIS	Placer order #
199406100600	Date/Time of transaction

OBR - OBR||123456^HIS||ART BLOOD||||||||||73123^WELBY^MARCUS|

Where:

123456^HIS	Placer order #
73123^WELBY^	Attending physician DOC_NUM^LAST^FIRST
ART BLOOD	Test Code being ordered (Universal Service ID)

Example Transactions

Receiving Order

```
MSH|^~\&|HIS|HOST|||199406100800||ORM|5901|P|2.1|
PID|||115401||JONES^FRED^Q||19631027|M||C
PV1||I|3E^305^01|R|||73123^WELBY^MARCUS
ORC|NW|89003^HOST|||||||199406100745|
OBR||89003^HOST||ARTERIAL
BLOOD|||||||||||73123^WELBY^MARCUS|
```

The HOST system is sending a new order for ARTERIAL BLOOD test to the ancillary system. The receiving system would send the following message if the message was received correctly:

```
MSH|^~\&|RESP|RESP|||199406100801||ACK|0001|P|2.1|
MSA|AA|5901
```

The message control id of '5901' sent in the order is placed in the MSA segment of the acknowledgment message.

Sending Results

```
MSH|^~\&|RESP|RESP|||199406100900||ORU^R01|0002|P|2.1|
PID|||115401||JONES^FRED^Q||19631027|M||C
ORC|RE|89003^HOST|||||||199406100855|
OBR|1|123456^HIS||ART BLOOD|||19951108132900|19951108132
900||JM||||19951108132900|RRadial^^^|||||||1995110813290
0|||||||||||SF|SF
OBX|0|ST|AnalyzeDate|1|1998-08-06 14:34:00.000000
OBX|1|ST|DrawDate|1|1998-08-06
OBX|2|ST|PH|1|7.363||7.35-7.45|
OBX|3|ST|PCO2|1|61.8|mmHg|35-45|A
OBX|4|ST|PO2|1|68.5|mmHg|80-100|D
OBX|5|ST|HCO3A|1|32.3|mmol/L|22-26|A
OBX|6|ST|BE|1|11.0|mmol/L|-2-2|A
OBX|7|ST|THB|1|12.2|g/dl|12-18|
OBX|8|ST|O2HB|1|98.9|%|85-98|A
OBX|9|ST|FIO2|1|21.0|||
```

The instrument computer is sending the results to a sample. The order number is 89003, analyzed at 7:45am on 6/10/1994.

```
MSH|^~\&|HIS|HOST|||199406100801||ACK|0001|P|2.1|
MSA|AA|0002
```

The message control id of '0001' sent in the order is placed in the MSA segment of the acknowledgment message.