Digital DEC/EDI

Introduction

Revised for Software Version: Digital DEC/EDI Version 4.0

Compaq Computer Corporation
Houston, Texas

November 2001

Printed in the U.S.A.

Contents

Preface

Chapter 1 What Is Electronic Data Interchange?

Chapter 2 What Is Digital DEC/EDI?

Chapter 3 Document Flows in Digital DEC/EDI

Purpose of this book

The purpose of this book is to provide an overview of Digital's system for **Electronic Data Interchange** (EDI), which is called Digital DEC/EDI.

The book begins with a general introduction to EDI. Subsequent sections of the book deal with the main components of the Digital DEC/EDI system, and how business documents are processed and communicated to trading partners.

Readership

This book is intended for anyone who needs to find out the basic principles of EDI as a way of doing business, and the overall structure of a Digital DEC/EDI system in particular. The book introduces the key features of the system and explains the terminology it uses.

If your organization is about to set up a Digital DEC/EDI system, then this book will give you an understanding of the whole system, before you approach the other more detailed technical books in the set.

Finding your way around this book

This is a narrative book that should be read right through. To use this book as a reference manual, there is a comprehensive contents listing and an alphabetically sorted index. The Glossary explains both EDI specific and generic terminology that you will come across in this book.

Related Books

This is one of a set of Digital DEC/EDI books. The complete list is as follows:

- *Digital DEC/EDI: Introduction*

  This book introduces general EDI concepts and Digital's EDI system, Digital DEC/EDI. It describes the main components of the Digital DEC/EDI system, and how business documents are processed and

communicated to trading partners. The book seeks to establish the concepts and terms used by Digital DEC/EDI. These are also summarized in a glossary.

You are strongly recommended to become familiar with the material in this book before proceeding to install or use Digital DEC/EDI.

- *Digital DEC/EDI: Installation*

  This book describes how to install the Digital DEC/EDI software, how to perform basic system configuration and how to verify such an installation. It describes how to install the Application Client, Server, Cockpit and CommandCenter components.

- *Digital DEC/EDI: Application Development*

  This book describes the Application Client interfaces and the means of connecting business applications to the Application Client. It also details the creation and deployment of mapping tables as part of the process of integrating applications with Digital DEC/EDI.

- *Digital DEC/EDI: User's Guides (Tru64UNIX and Open VMS)*

  These guides contain information on setting up and operating Digital DEC/EDI systems. They also contain information covering configuration, maintenance and problem solving.

  The term User's Guides is used throughout this book to refer to the following books which are provided along with the Digital DEC/EDI Server they describe.

Tru64 UNIX	Digital DEC/EDI: tru64 UNIX User Support Manual
OpenVMS	Digital DEC/EDI: OpenVMS User Support Manual - Volume 1
	Digital DEC/EDI: OpenVMS User Support Manual - Volume 2

- Release Notes

  Further to the above, each software kit contains a set of release notes applicable to that software. These release notes contain information about known product problems (with workarounds where appropriate) and any operational tips or hints not provided as part of the above documentation set. You are strongly recommended to review these release notes before installing the software. Refer to the appropriate installation guide for information on how to locate the release notes.

- On-line Documentation

  Comprehensive on-line documentation is supplied with the Digital DEC/EDI software: for example, on-line help libraries and UNIX man page help information. In addition the Digital DEC/EDI Cockpit and CommandCenter kits contains the *Digital DEC/EDI: Error Messages Help Library.* This contains all error messages the product may log along with a description of why the message occurred and what to do about it. It is provided in MS-Windows help library format. In addition, the CommandCenter CD-ROM provides on-line versions of all Digital DEC/EDI books in Adobe Acrobat format.

Digital DEC/EDI InfoCenter

For further information on Digital's EDI and Electronic Commerce Solutions and Services, please visit the EDI InfoCenter on the World Wide Web. The location is:

```
http://www.decedi.com
http://www.compaq.com/edi/
```

Related Third Party Documentation

Refer to the documentation provided with third-party products for installation and configuration details.

Typographical conventions

Digital DEC/EDI / Compaq DEC/ EDI

The ownership of DEC/EDI was transferred to Digital GlobalSoft Ltd, a subsidiary of Compaq Computer Corporation based in India with effect from May 1, 2001. Consequent to this transfer, the name of the product was changed to Digital DEC/EDI. There may be references made to the existing name of the product, Compaq DEC/EDI in various sections of the documentation and screen display. We are in the process of implementing the name change across the product code and documentation. This is expected to be completed within the next couple of months. Pending the completion of this, all references to Compaq DEC/EDI in the documentation pertain to the Digital DEC/EDI product. Please refer to the product website at `www.decedi.com` for further information on the transfer of ownership.

Chapter 1 What Is Electronic Data Interchange?

This chapter answers the general question, “What is Electronic Data Interchange?”, and describes some of the benefits it provides in comparison with traditional paper-based trading.

What is Electronic Data Interchange (EDI)?

EDI is the electronic transfer of structured business data (for example, a purchase order) from one business application to another. This implies that an EDI system needs to provide three main things:

- Application Integration

  The means to receive data from and supply data to suitably authorized business applications.

- Common Formats

  The means to ensure EDI data to be exchanged with trading partners complies with defined formats for such data.

- Communications

  The means to transfer data to other EDI systems by using agreed communications protocols.

Strictly, EDI implies the exchange of data between business applications and not between people. While it is common to refer to “doing EDI with trading partner X”, it is in fact the business applications that the trading partner is running that are “doing EDI”.

Throughout this book where reference is made to “trading partners” it is referring to the EDI systems and business applications they are running. Note the contrast here between EDI and the exchange of electronic mail.

The latter is characterized by the exchange of mainly unstructured information between people.

Implementing EDI

A successful EDI implementation involves close co-operation between the sender and receiver of business documents so that data in the document means the same to both. Businesses, companies or organizations that transfer documents by using EDI are known as **trading partners**. Each type of document transferred, such as a purchase order or an invoice, must be the subject of an individual **trading partner agreement**.

Trading partner agreements identify and specify all EDI specific elements of a document and the way in which it is to be transmitted. These elements include data-field characteristics, document type, communications link details, and application programs that send and receive the documents.

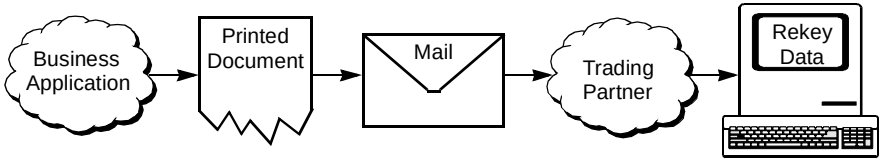
EDI enables your company to exchange documents with many trading partners; these can be external companies, or internal organizations within your own company. Each pair of trading partners may have several agreements between them, enabling them to conduct different types of business transaction, by using different document types for each one.

EDI is platform-independent. The computers that different trading partners use do not have to be from the same manufacturer: nor do different trading partners have to have the same internal business methods or internal document formats. All that matters is that their interaction with one another adheres to the trading partner agreement: the main components of this agreement are the document types and the communications link.

Traditional Document Exchange

At its simplest level, EDI replaces the traditional paper-based document transfer process that entails manual intervention at several stages, as outlined in Figure 1-1 *Paper-based Document Transfer*.

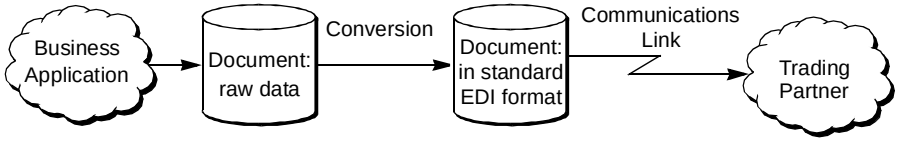
Figure 1-1 Paper-based Document Transfer



Electronic Document Exchange

By using an electronic communications link between computers (see Figure 1-2 *EDI Document Transfer*), EDI enables computer applications to "talk" directly to each other. For example, outgoing invoices would be routed to an accounting application at the receiving end. This method improves the security of data by reducing manual intervention, eliminates the need for re-typing, error correction and improves business cycle times.

Figure 1-2 EDI Document Transfer



Standard Document Types

All documents exchanged between trading partners using EDI must conform to mutually accepted EDI standards. It follows that multi-national trading partners operating within this framework need to adopt and conform to internationally agreed standards.

EDI standards apply broadly on two levels:

- The type of document to be exchanged

  For example, an invoice or a purchase order. Organizations such as the UN and ANSI maintain the standards relating to the type of document.

- The internal elements which make up that document

  For example, addressee, destination, and the informational elements that the document is designed to convey. These elements are composed in an agreed syntax such as EDIFACT, TRADACOMS or X12: there are many others. They are created and administered by bodies such as the **International Standards Organization** (ISO), **American National Standards Institute** (ANSI) or other EDI standards bodies.

An EDI standard defines a set of named documents (for example, EDIFACT defines document types for price quotations (QUOTES), purchase orders (ORDERS), invoices (INVOIC) and so on) and their constituent parts.

Standard Document Structures

Standards apply not only to the type of document, but also to the structure and layout of its constituent parts.

The parts of a document fit into a defined hierarchy and they are referred to collectively as a **dictionary**. At the highest level in the hierarchy are **segments**. Segments are composed of **elements**. These elements may have either a simple or a composite structure depending on whether they contain a single item of data, or multiple items of related data.

An example segment might contain the name and address of the sender of a particular document. Individual elements would contain the different parts of that address.

EDI Document Definitions

An EDI document definition is normally composed of data organized into three areas:

- **Header area**

  The Header area includes addressing information and other data relevant to the document as a whole.

- **Detail area**

  The Detail area typically comprises a list of line items. For example, the Detail area of a purchase order might list the separate items to be purchased: “6 of left-handed widgets, 2 packs of shiny washers, 3 tins of striped paint”, and so on.

- **Summary area**

  Summary area presents a set of totals resulting from the calculations set up in the Detail area. In a purchase order, for example, this would include the total amount of spending on each item listed in the Detail area, and the total value of the purchase order.

This partitioning into areas reflects the structure of most business documents.

Within an area, related segments can be grouped together in **loops**. This would be used, for example, to group together all addressing information about one trading partner to make it distinct from the information about another trading partner. More specifically, and for greater efficiency in communication and transaction, addressing information about any trading partner can contain more than one address. For example, the delivery address on a purchase order and the corresponding invoice address might be different.

Business document definitions are designed to encompass a wide range of potential applications across many industries. In practice, this often means individual definitions are very comprehensive because the standards bodies like the ISO and the UN are influenced to design the format to satisfy the needs of many companies working in many industries.

By design, the definitions form a super-set of all the various requirements, and many pieces of a document are not **mandatory**. Mandatory data is that minimum amount of the document type considered to be vital to any usage of that document. Other parts are **optional** and some pieces are **conditional**, that is, their presence may depend on the presence (or otherwise) of other pieces of the document. Each trading partner agreement needs to determine exactly which parts of a document definition are exchanged.

EDI documents destined for the same recipient (as defined by the applicable trading partner agreement), may be aggregated into an **interchange**. If there is more than one type of document in an interchange, all documents in each type can be grouped into a separate **functional group**. For example, all invoices might be included in a particular functional group: all purchase orders could be in another.

Non-standard Data Interchange

In general, EDI - Electronic Data Interchange - platforms are not limited to the exchange of data that complies only with EDI standards. EDI communications links can also be used to convey other forms of electronic data, such as computer-aided design and manufacturing data (CAD/CAM).

Standard Communication Links

As well as converting documents to and from standard formats, an EDI system provides communication links to trading partners. To ensure that a file is never lost, and that it is correctly recognized and processed at the destination, there are industry standards governing the way that files or documents are transmitted.

A set of communication standards is called a **protocol**. The two communicating computer systems need to be able to acknowledge one another, and to conduct a recognized transaction between themselves. This is analogous to the conventions governing the mailing of letters: you need to stamp and address the envelope in a certain way, so that the mail service can recognize and deliver the letter, and you need to put the sender's address on the reverse if you wish to receive notification of non-delivery.

There are various transport methods for delivering files from one computer system to another:

- Direct connection
- Value Added Network (VAN)
- Magnetic tape or disk

The following sections describe them briefly.

Direct Connection

You can establish a **direct connection** with a trading partner over a leased telecommunications line. When the two computer systems have acknowledged one another, your system can send the EDI data, which is immediately received by the destination system.

The choice of communications protocol is part of the trading partner agreement. There are several different communications protocols in use to transfer EDI data.

Value Added Network (VAN)

You can send an EDI document file over a **Value Added Network** (VAN). A VAN is a third-party agency that takes responsibility for delivering files between trading partners. Instead of establishing a connection with each trading partner, your computer system connects to the VAN, which then routes files to the appropriate partners according to routing data accompanying the file.

The VAN is able to pass files between any pair of companies that subscribe to it, and it may offer to provide a number of additional services. Typically, once the VAN has accepted a file from a sender, it guarantees to deliver the file, or if it is unable to do so after repeated attempts, to return a non-delivery report to the sender. The VAN also provides a queuing service, so that the sender and recipient do not have to have communications jobs scheduled for the same time. The sender transmits the file to the VAN according to its own schedule, and the recipient picks up the file from the VAN according to its schedule. The VAN may perform other functions, such as validation and authorization checks.

There are many VANs available, providing this kind of service to subscribers. Each VAN prescribes a particular protocol: different VANs prescribe different protocols. The protocol governs how your computer system establishes a connection to the VAN, exchanges information with the VAN (for example sending a file, or receiving a delivery report) and quits the connection to the VAN.

Magnetic Tape or Disk

You may elect to send EDI data to a trading partner by recording it on disk or tape and physically delivering it. This method is simpler than using a direct connection or a VAN: it is less rapid but more secure.

This method can be useful either for transmitting large volumes of data where speed is not important, and the expense of electronic communication is not warranted or for testing purposes when you are developing agreements with a trading partner. The method is also useful for high-security transactions, as it is less easily susceptible to attack by computer hackers. For this reason, banking documents are often delivered on tape or disk.

Comparing EDI With Paper-Based Trading

The following table compares the relative merits of electronic trading and traditional paper-based trading.

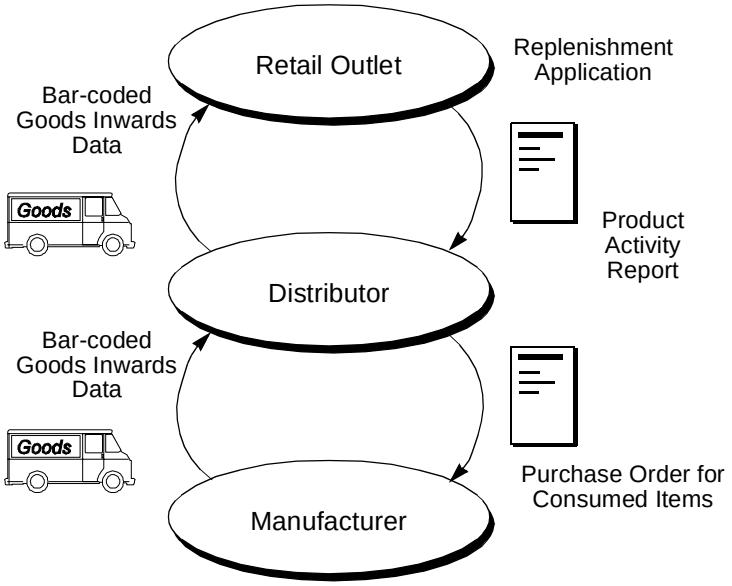
Table 1-1 Comparing EDI With Paper-Based Trading

Commercial Area of Interest	EDI	Paper
Time in transit	Less	More
Security of delivery	More	Less
Security from tampering/hacking	More	Less
Security from human error	More	Less
Resources - storage space	Less	More
Resources - staff	Less	More
Resources - staff accommodation	Less	More

Integrated EDI

When EDI is fully integrated into an enterprise, all transactions from point of sale or use, back up the supply chain to the point of manufacture can be linked by computer application and EDI. Figure 1-3 *Integrated EDI* shows an example.

Figure 1-3 Integrated EDI



In this simplified example, consumer demand at a Retail Outlet is assessed by an application program, that issues replenishment requests to a Distributor as stock levels reach some planned minimum.

The Distributor checks their stock, and if necessary, issues purchase orders to a Manufacturer.

When the goods are manufactured they are shipped to the Distributor and bar-coded through Goods Inwards. The Distributor now knows that stock levels of the product are satisfactory.

The Distributor ships the goods to the Retail Outlet, where they are bar-coded into Goods Inwards, and placed on shelves ready for sale.

When the data relating to the goods is available in electronic form from either Electronic-Point-Of-Sale (EPOS) machines or from bar-code readers, it can be manipulated by business application programs, and transported by EDI to the people and systems that need to know.

Summary of EDI

To summarize this chapter, EDI implies the exchange of structured business data between computer applications. In practice this means:

- Information is transferred electronically, rather than on paper. This eliminates the need to enter the data manually into the destination computer. The same raw data can serve as the basis for more than one document: for example, the purchase order you receive, and the delivery advice you dispatch with the ordered goods.
- Transactions between trading partners are governed by a trading partner agreement.
- An agreement specifies the standard document types to be used for each type of transaction. The EDI system is responsible for converting information to and from the standard document formats.
- An agreement also specifies the means of communication to be used. The EDI system is responsible for sending and receiving document files according to a standard communications protocol.

By using EDI, each trading partner is free to organize its work as it wishes, and EDI ensures that information is communicated in a form that other trading partners can recognize and handle.

EDI is a growing method of doing business. Many organizations insist, as a condition of tender, that their trading partners use EDI. This provides many significant business benefits, including:

- Marketing competitiveness
- Administrative cost savings
- Shorter time to market
- Better quality control
- Improved corporate trading relationships

The remaining chapters of this book describe Digital's EDI system, Digital DEC/EDI.

Chapter 2 **What Is Digital DEC/EDI?**

This chapter provides a high level look at the main components of Digital's EDI system, Digital DEC/EDI.

The Digital DEC/EDI System

A minimal Digital DEC/EDI system is comprised of the following main components:

- Digital DEC/EDI Server
- Digital DEC/EDI Application Client
- Digital DEC/EDI User Interface

The Digital DEC/EDI User Interface is used to configure, maintain and monitor the Digital DEC/EDI Server, and to integrate business applications with Digital DEC/EDI. The following user interfaces are available, depending on the Digital DEC/EDI Server platform in use:

- Digital DEC/EDI Cockpit

  This PC-based monitoring application is required for use in conjunction with the Digital DEC/EDI Server for Tru64 UNIX, and may also be used in conjunction with the Digital DEC/EDI Server for OpenVMS.

- Digital DEC/EDI CommandCenter

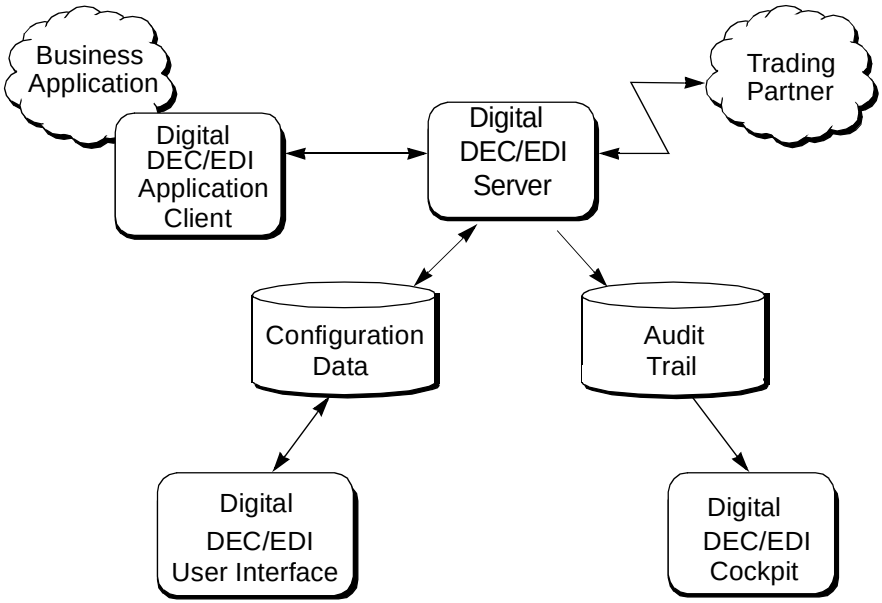
  This PC-based suite of applications which includes the Digital DEC/EDI Cockpit, is required for use in conjunction with the Digital DEC/EDI Server for Tru64 UNIX. A limited set of functionality that enables integration of business applications is available for use in conjunction with the Digital DEC/EDI Server for OpenVMS.

- INTERCHANGE

  This OpenVMS based user interface is provided with the Digital DEC/EDI Server for OpenVMS, and is used to configure, maintain and monitor the Digital DEC/EDI Server. In addition, the Digital DEC/EDI Cockpit may be used to monitor the Digital DEC/EDI Server.

The components may be combined to create a Digital DEC/EDI system, depicted, for example in Figure 2-1 *Digital DEC/EDI Systems Components*. The following sections describe each one:

Figure 2-1 Digital DEC/EDI Systems Components



Digital DEC/EDI Server

The Server is the main functional component of the Digital DEC/EDI system handling the exchange of EDI data between business applications and external trading partners.

A configuration database defines what operations the Server is required to perform. This includes what business applications may use the system, what data they may exchange with which trading partners, and how the data is to be transmitted.

A full audit trail is maintained of data processed by the Server. This may be used either to see what documents are currently being processed, or to look at data that has completed processing.

Digital DEC/EDI Application Client

Business applications use the EDI services provided by the Server through the Digital DEC/EDI Application Client. The Application Client permits business applications to post and fetch files of business data. Also, an application can track business data within the Server, both as a means of monitoring work in progress, and as a means to producing historical business reports.

Configuration Database

The Digital DEC/EDI configuration database defines what operations the Server is required to perform. This includes the following:

- Business Applications

  The names assigned to the business applications that are registered to use the Digital DEC/EDI system, and where they are located.
- Trading Partner agreements

  The trading partners with which you exchange data, and what EDI standards and communications methods you wish to use.
- EDI Document definitions

  The EDI dictionaries and document definitions. These may be customized for generic or trading partner specific variations.
- Communications details

  Details of the protocols and communications parameters.

For Tru64 UNIX Servers, the CommandCenter is used to configure and maintain this data.

For OpenVMS Servers, the INTERCHANGE interface is used to configure and manage this data.

Audit Trail

Digital DEC/EDI maintains an audit trail of the data processed by the Server.

This may be used either to view the documents that are currently being processed, or to examine data that has completed processing.

In many instances, the ability to audit and manage this business data is a legal requirement. This information gives visibility of the data processed by the system, and can also be used to identify and track exceptions.

Tracking data is archived and can be examined at a later date if necessary. Failed data can be reset and re-sent if requested, for example if it has been lost outside of the scope of the Digital DEC/EDI system.

Activity Log

The Digital DEC/EDI system maintains a log of key events and exceptions that occur, for example, system start-up, shutdown or component failures. This record is called the Error Log and can be accessed when required.

Digital DEC/EDI User Interfaces

Digital DEC/EDI Cockpit

The Cockpit is a PC-based graphical user interface that allows the flow of EDI documents within a Digital DEC/EDI Server to be monitored in real-time. Any documents that fail during processing are identified easily, allowing the cause of the failure to be investigated directly. A limited set of control functions enable re-starting or cancelling of failed document transfer.

The Cockpit also includes the ability to control access to particular elements of the data within the Digital DEC/EDI audit trail.

Digital DEC/EDI CommandCenter

The CommandCenter is a PC-based graphical user interface that is used to configure the Digital DEC/EDI system. The CommandCenter, which includes the Cockpit, is provided as a set of PC-based applications.

For OpenVMS and Tru64 UNIX Servers, the CommandCenter enables the development of Mapping Tables for integrating business applications with Digital DEC/EDI.

For Tru64 UNIX Servers, the CommandCenter provides additional applications which are required for creating and maintaining the configuration data for the Digital DEC/EDI system. For example, this

includes the trading partner profiles, that is, information about which EDI documents the system expects to exchange with which trading partners.

INTERCHANGE

The INTERCHANGE user interface is only available for the Digital DEC/EDI Server for OpenVMS. This terminal based interface is used for creating and maintaining the necessary configuration data for Digital DEC/EDI, as well as for monitoring the flow of EDI documents within the Digital DEC/EDI system.

Practicalities of a Digital DEC/EDI System

Digital DEC/EDI provides an extensive range of configuration options, covering various hardware and software combinations.

OpenVMS Server

A minimal Digital DEC/EDI system based around an OpenVMS server must contain at least one of each of the following components:

- Digital DEC/EDI Server

  This includes the INTERCHANGE user interface.
- Digital DEC/EDI Application Client

  This may be deployed on the same node as the Digital DEC/EDI Server, or on additional nodes within the same network.
- Digital DEC/EDI Command Center

  This interface is to enable integration of Applications.

Tru64 UNIX Server

A minimal Digital DEC/EDI system based around a Tru64 UNIX server must contain at least one of each of the following components:

- Digital DEC/EDI Server

  This requires the Digital DEC/EDI CommandCenter to be deployed on a PC within the same network.
- Digital DEC/EDI Application Client

  This may be deployed on the same node as the Digital DEC/EDI Server, or on additional nodes within the same network.
- Digital DEC/EDI CommandCenter

  This includes a copy of the Digital DEC/EDI Cockpit, plus additional applications to configure and maintain the Digital DEC/EDI Server, and to integrate business applications with Digital DEC/EDI.

Application Client

The Application Client runs on the same node as the business application, although this may be different to the node on which the Server is running. Many different applications may use the Digital DEC/EDI system concurrently, and where they are on different nodes to the Server, each node requires a copy of the Application Client.

Cockpit and CommandCenter

Depending on the number of people you wish to have access to the Digital DEC/EDI system, you may wish to deploy additional copies of the Cockpit and CommandCenter. Each separate PC that needs to access the system needs its own copies of the Cockpit and CommandCenter.

In practice, you may provide only a small number of users with copies of the CommandCenter (that is the ability to view or modify the system configuration) whereas you could provide a larger number of people with the Cockpit (that is the ability to monitor EDI data in the system).

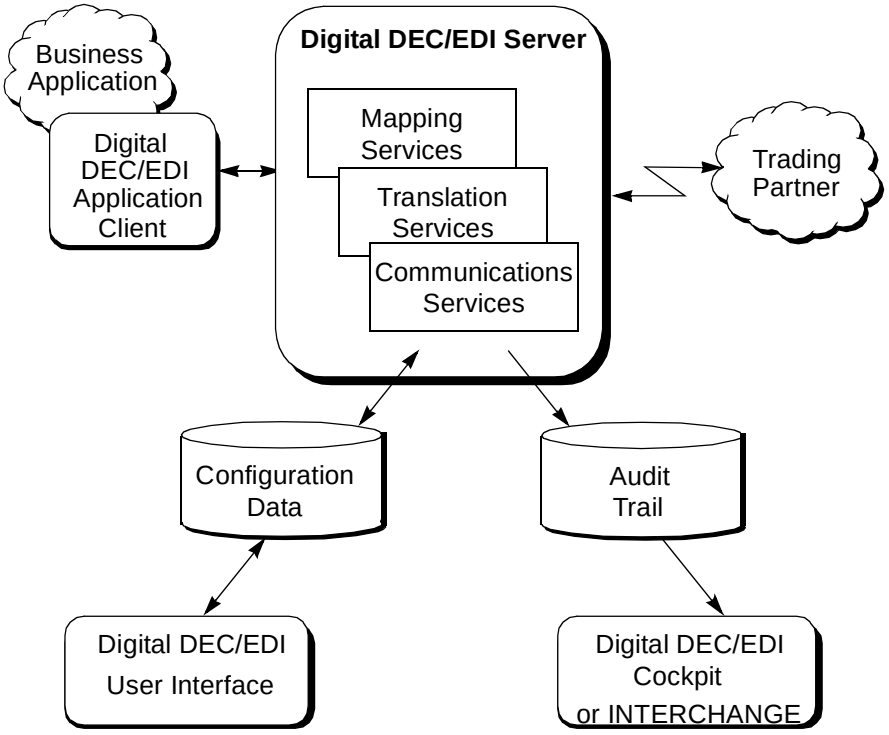
It is possible for the Cockpit and CommandCenter to connect to more than one Server, and to connect to both OpenVMS and Tru64 UNIX Servers.

On a point of terminology, Digital DEC/EDI refers to the direction of data as being either ***outbound*** *or* ***inbound****. Outbound data is that which is "posted" by a business application to the Server via an Application Client. The data is finally sent to a trading partner. Inbound data enters the Server from a trading partner. A business application "fetches" inbound data from the Server via an Application Client.*

The Server in More Detail

Figure 2-2 *Digital DEC/EDI Server Components* shows a more detailed view of the three main services applied by the Digital DEC/EDI Server.

Figure 2-2 Digital DEC/EDI Server Components



These services are:

- Mapping Services
- Translation Services
- Communications Services

For the majority of installations, data is processed by each of the three services in turn: for outbound data, this means mapping, translation and then communications. For inbound data, the order is reversed. There are other options for routing data through the server. These are described later in this book (See Section *Alternative Document Flows in Digital DEC/EDI* on page 3-7).

Mapping Services

The Mapping Services provide the link between the data format as supported by each different business application, and that defined by the EDI standards bodies. At the simplest level, this might imply simple re-ordering of data. More typically, it also involves calculation of some values and conversion of data between data types.

The Mapping Services operate by using rules that you specify in a Mapping Table. The creation of Mapping Tables is a fundamental part of the configuration of Digital DEC/EDI. Normally each business application requires the definition of at least one Mapping Table. A Mapping Table comprises a detailed description of the format of a particular application file, the definition of the corresponding EDI documents, and rules for mapping each element of data between the two.

In addition to the simple data mapping services (for example “create destination field y from source field x”) Digital DEC/EDI also provides many other powerful mapping features including:

- Data manipulation

  Individual mapping statements are constructed by using a rich mapping language that allows complex expressions to be created of the form *destination_data_field = mapping_expression*. A mapping expression typically combines elements of source data using predefined functions, simple mathematical expressions, and constant or computed values.

- Data type conversion

  The ability to process sections of application file data of many different types (for example: textual, integer numeric, real numeric, and date/time).

- Hook routines

  The ability to customize individual Mapping Tables by coupling user-defined processing functions at any stage in the mapping process. This can be used, for example, to perform some data manipulation not directly supported within the mapper, or to fetch data from an external database.

- Handling of variants

  The Mapping Services support conditional mapping. For example, different trading partners may request or require minor differences in what they send or receive. The mapper is able to support this.

- Support for outbound batching

  This facility is particularly useful for companies that need to transmit large numbers of documents.

- Support for inbound batching

  This facility enables all documents within one user-definable “batch” (typically a single transmission file) to be fetched into one application file. Mapping expressions enable the user to specify what constitutes a batch, and to determine whether to continue fetching documents.

- Support for business data references

  This facility enables any data value to be stored in one of five business reference fields within the Digital DEC/EDI audit trail. These business reference fields may then be used to simplify tracking and extraction of data from documents within the Digital DEC/EDI system.

Mapping tables are created interactively by using the Mapping Table Editor (See *CommandCenter* on page 2-15).

However they must be converted into a different form before they can be used by the Mapping Service within the Server. This is an automated process referred to as **compilation**. All Mapping Tables must be compiled prior to use. The compilation process does two things. Firstly it checks, amongst other things, that the Mapping Table expressions you have entered are valid. Secondly it produces a more compact version of a Mapping Table that allows the Mapping Services to run more effectively when processing EDI data.

Translation Services

The Translation Services perform the conversion to or translation from data formatted to conform with published EDI standards. The EDI standards bodies define the content and syntax of EDI documents, such as purchase orders. In processing outbound data, the Translation Services **convert** mapped application data to the EDI standard form. In processing inbound data, the Translation Services **translate** data from the EDI standard format to that required for mapping to the application file.

Digital DEC/EDI Version 4.0 enables you to exchange documents that conform to the following EDI standards:

- X12/TDCC

  This international standard defines commonly used business transactions in a formal, structured manner called transaction sets. This standard supports composite elements and extended conditional requirements.

- TRADACOMS

  This standard supports the transmission of many documents in one transmission file.

  The rules were developed from the SITPRO syntax rules which were developed for international trade applications.

- EDIFACT

  This is a world-wide standard, largely used for international trading. It embraces UN standards for document format, and ISO standards for envelope format.

- ODETTE

  This is a world-wide standard, largely used in the motor manufacturing industry. It embraces ISO standards for envelope format.

  ODETTE is related to EDIFACT (they share a common syntax for the construction of EDI document definitions), and both standards are handled by the same converter and translator on the Translation Service; for this reason, the Digital DEC/EDI books often refer to EDIFACT/ODETTE documents.

For all the above standards, tables of document types are supplied as part of the Digital DEC/EDI software (See*Message Update Services* on page 2-16). You can modify these document types, so as to conform to any other industry standards, such as:

- EANCOM - International Article Numbering Association Standard
- CEFIC - European Chemical Industry Standard
- EDIFICE - European Electronics Standard

These three standards are based on EDIFACT. In general, many industries adopt particular variations or subsets of prominent international standards. For example, VICS is based on X12.

In addition, you can apply variations to the standard document types at the level of individual trading partners. These are called **Trading Partner**

Specific documents. Many trading partners can share a defined **Trading Group**.

Communications Services

The Communications Services support the physical transfer of EDI data between Digital DEC/EDI and trading partners.

There are various standard methods of handling communications with trading partners. Each method has its protocol, defining a strategy of exchanging messages and acknowledgments with a trading partner, so that the sender can be sure that the message has been properly received and interpreted by the trading partner.

The Digital DEC/EDI Communication Services provide the following **gateways** for exchanging EDI data with trading partners:

- Bisynchronous communications

  Via the **CLEO 3780** gateway. **Scripts** for connections to GEIS EDI*Express and BT Tymnet are also provided. Your local Digital DEC/EDI Service supplier will be able to provide you with additional scripts other than those provided with the Digital DEC/EDI software.

- Import/Export gateway

  For exporting files to, and importing files from disk. This would prove a useful option for connecting to a proprietary communications system not directly supported by Digital DEC/EDI, or for use during preliminary system testing or pilot operation, where communications links to external trading partners are not established.

- Pedi gateway

  For store-and-forward connection to a trading partner, conforming to X.400 standards. This gateway supports the transmission and receipt of the following X.400 messages types:

  – 1984 P0 and P2

  – 1988 P0 and P2

  – Pedi (X.435)

- OFTP gateway (ODETTE File Transfer Protocol)

  For direct real-time connection to a trading partner or Value Added Network using the X.25 network.

- SMTP/MIME gateway

  This gateway is only available for Digital DEC/EDI for Tru64 UNIX.

  Used for sending and receiving via the Internet electronic mail (e-mail). SMTP is the abbreviation for Simple Mail Transfer Protocol: MIME is the abbreviation for Multi-purpose Internet Mail Extensions.

The Application Client in More Detail

The Digital DEC/EDI Application Client provides functions that enable files of application data to be posted to or fetched from the Digital DEC/EDI Server, or tracked within the Server. These functions are provided through both command line and application programming interfaces.

Interfacing business applications to Digital DEC/EDI is described in more detail in Chapter-4 *Connecting Applications to DEC/EDI*.

Cockpit

The Digital DEC/EDI Cockpit is a PC-based application that runs under MS-Windows. It provides the user of the PC with the means to observe the processing of EDI data within a Digital DEC/EDI Server, and reset any failed documents. The design of its interface is typical of the current generation of PC-based office applications. All operations can be performed by using pull down menus, while the more commonly used operations can be more easily invoked by using toolbar icons or “point-and-click” operations. Extensive help facilities are provided as part of the application.

The main features provided by the Cockpit include:

- Summary Display

  The ability to see at a glance the status of EDI data being processed by a given Server. A key feature of the display is the categorization of data within the Server in terms of its state of processing. That is, the display shows how many EDI documents are in which of the many possible

different processing states. This provides early indication of possible system bottlenecks, and clear indication of any data that may fail within the system.

From the summary display, it is possible to connect to more detailed displays of data in the system. For example, if any data fails, it is possible to display all the detailed information about the failure by using simple “point-and-click” operations.

The display may be configured to be updated automatically on a periodic basis or updated on demand. When configured to be updated automatically, the display provides a view of the state of data within a Server in real-time.

- Detail Displays

  In addition to the summary display, several detail displays are provided. These detail displays together provide access to all information maintained by the Server about EDI data in the system. This includes the ability to see all audit trail entries for specified EDI documents, and the ability to see the contents of the EDI documents themselves.

- Customizable Displays

  All the main displays used by the Cockpit are customizable in respect of the amount of data being shown, and the representation of that data (font, color). This means, for example, where a display is used to view audit trail information, you can select which individual items of audit trail information you wish to see. Such selections may be stored as **views** and applied to detail displays as needed. For example, for normal successful transactions, you may wish to see only the time the data entered the system and the time it completed processing. Whereas for failed transactions you may wish to see far more detailed information: who sent it, what sort of data it is, when did it fail, and so on. You can do this easily by using appropriately defined views.

- Restarting Failed Documents

  Data may fail processing within the Server. Typically this occurs where there is a mismatch between the EDI data being processed and that which the Server is configured to expect. For example, a particular document being processed may contain too much or too little data. The Cockpit provides an option to reset failed documents, and provides access to the information needed to determine the cause of failures. Once the reasons for failure have been determined, and corrected, the Cockpit is used, either to restart, or to cancel any further processing of that data.

- View the Digital DEC/EDI Error Log

  The Digital DEC/EDI Error Log contains error and other messages that record events that happen on the Server. The Cockpit allows this to be viewed.

- Multiple Server Capability

  Many Digital DEC/EDI installations may have only one Server. However where more than one Server is being used, the Cockpit is able to connect to multiple Servers concurrently. This means that from a single PC, it is possible to see the state of many Servers.

- Access Controls

  The Cockpit includes the Access Control Editor, which enables user access controls to be defined.

  This enables control of access to the data that may be viewed or managed by a particular Cockpit user. This specifies whether the user may:

  – View business data only (read-only).

  – View and modify business data (reset, resend, cancel batch, archive mapper events).

  – View and modify data for a restricted set of applications and/or trading partners.

  – View data for mapper events, documents, batches or transmission files.

  – Use the CommandCenter applications

- Business Data Views

  A number of user definable fields are provided in the Digital DEC/EDI Audit Trail, which may be used to capture business specific data during the processing of a document. This might include a purchase order number, or a supplier code, or any combination of fields within the document.

  The Cockpit Summary or Display Views may filter documents based on these fields, and allow business specific names to be assigned to the fields displayed.

  This powerful feature enables you to quickly locate and monitor documents based on the key information that is important to your business.

CommandCenter

Shown below are the icons used for each of the different editors.

Like the Digital DEC/EDI Cockpit, the Digital DEC/EDI CommandCenter is PC-based. It is comprised of a set of editors that are used to create and maintain configuration information for a Digital DEC/EDI Server. This configuration information splits into five main pieces, and separate editors within the CommandCenter are used to address each piece:

- Management Services Editor

  May only be used with Tru64 UNIX Servers.
  Allows you to register details about which business applications may use a given Server and on which nodes they exist, and which Translation and Communications Services a Server should provide.

- EDI Table Editor

  May only be used with Tru64 UNIX Servers.
  Allows you to define the structure and content of individual EDI documents to be exchanged with trading partners. This information is stored in EDI Document Tables.

- Mapping Table Editor

  May be used to develop Mapping Tables for both OpenVMS and Tru64 UNIX Servers.

  Allows you to define how the format of data required by a business application maps to the format of data implied by the selected EDI standard. The editor allows you to produce compiled versions of Mapping Tables for use by the Mapping Services.

  The Mapping Table Editor allows you to specify values to be assigned to business data reference fields, for use with business data views.

- Communications Editor

  May only be used with Tru64 UNIX Servers.

  Allows you to define which communications options are being used and the parameters of individual communications links, either direct to trading partners or indirectly through third party networks. Communication Editor is also used to define cluster nodes.

- Trading Partner Editor

  May only be used with Tru64 UNIX Servers.
  Allows you to define the trading partners with whom the system will exchange data, the communications links to be used, the agreements that

govern which EDI documents are to be exchanged, and what enveloping (addressing) information is to be used. This information is held in Trading Partner Tables.

The style of the CommandCenter is very similar to that of the Cockpit. While it is designed to perform data entry tasks, the amount of information needed to be entered is minimized through the use of features such as cut, copy and paste, and drop-down lists where you select from pre-defined sets of options, and extensive ranges of defaults. Like the Cockpit, the CommandCenter may be used to configure and maintain more than one Server, and to transfer elements of configuration data between Servers.

Message Update Services

A Server is likely to be required to process many different document types from many different versions of EDI standards. While the CommandCenter provides the means to create and maintain the necessary EDI Document Tables, the time needed to enter all the necessary information that defines these document definitions and their constituent components is likely to be significant.

To simplify the configuration process, Digital DEC/EDI provides a **Message Updates Services** (MUS). This is a collection of tables that ships with Digital DEC/EDI. These tables contain the different supported versions of EDI standards as designated by the various EDI standards bodies.

Each version of the MUS contains a number of document definitions, and the dictionaries of segments, element and sub-elements from which the document definitions are constructed.

This means that when the appropriate MUS tables are loaded into the Digital DEC/EDI Server, the CommandCenter needs to be used only to create minor variations as required (as opposed to having to create all the EDI document definitions manually).

The MUS tables are re-shipped periodically as the EDI standards bodies define new versions of their standards.

Products and Platforms

The Digital DEC/EDI system components described in the previous sections are packaged into the following products:

- Digital DEC/EDI for Tru64 UNIX, Version 4.0

  This contains both the software for both the Application Client and Server. At installation time, you can choose whether to install either or both. The MUS tables are also shipped within this product.

- Digital DEC/EDI for OpenVMS Alpha, Version 4.0

  This contains both the software for both the Application Client and Server. At installation time, you can choose whether to install either or both. The MUS tables are also shipped within this product.

- Digital DEC/EDI Cockpit, Version 4.0

  This product contains the Digital DEC/EDI Cockpit. This includes software for both MS-Windows and Windows 95/NT platforms.

- Digital DEC/EDI CommandCenter, Version 4.0

  This product contains the Digital DEC/EDI CommandCenter, and a copy of the Digital DEC/EDI Cockpit. This includes software for both MS-Windows and Windows 95/NT platforms.

- DEC/EDI for Sun Solaris, Version 4.0.

  This product contains the DEC/EDI application client for the Sun Solaris platform.

- DEC/EDI for HP-UX, Version 4.0.

  This product contains the DEC/EDI application client for the HP-UX platform.

- DEC/EDI for RedHat Linux, Version 4.0.

  This product contains the DEC/EDI application client for RedHat Linux platform version 6.2.

- DEC/EDI for Windows NT , Version 4.0.

  This product contains the DEC/EDI application client for Windows NT with TRACK functionality.

Chapter 3 Document Flows in Digital DEC/EDI

This chapter describes the flow of business data through Digital DEC/EDI, outlining the main processing stages of that data flow. In addition, it introduces the different options for routing data through Digital DEC/EDI.

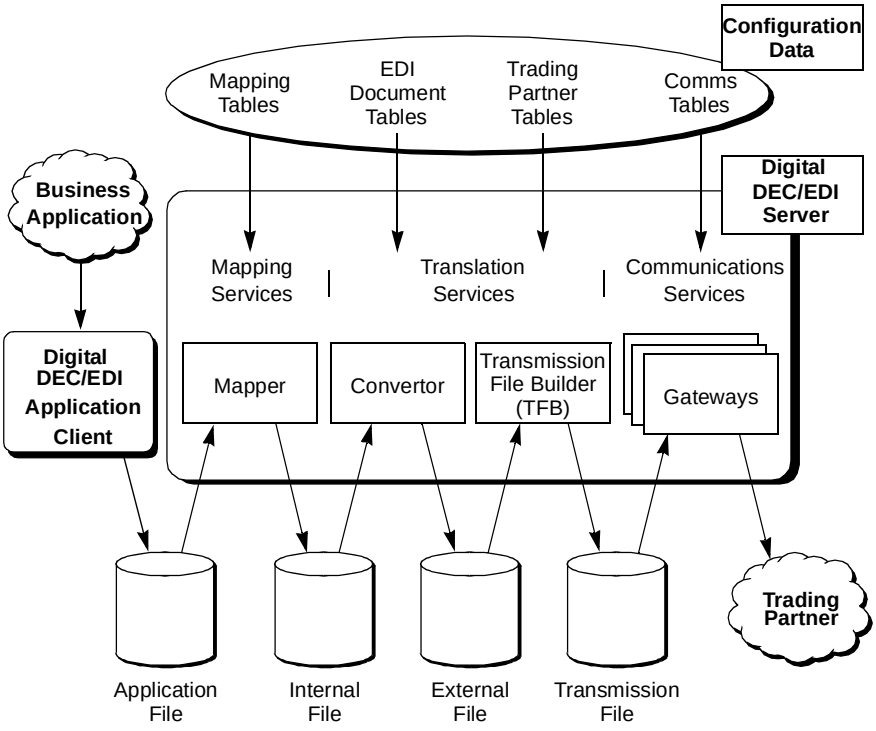
Digital DEC/EDI Data Flow

This section describes the flow of data through a Digital DEC/EDI system for both outbound and inbound data. This data flow applies to the majority of Digital DEC/EDI installations. There are other options for routing data through Digital DEC/EDI and these are discussed in Section *Alternative Document Flows in Digital DEC/EDI* on page 3-7. The data flow or routing to be used in a given instance is determined by how the Digital DEC/EDI system has been configured and what parameters are specified by the business application when invoking the Application Client.

Outbound Data Flow

This section describes the flow of outbound data through the Digital DEC/EDI system. This data flow is shown in Figure 3-1 *Outbound Data Flow*. The description introduces a number of new terms that are highlighted and explained in the text.

Figure 3-1 Outbound Data Flow



The following sections outline the steps involved:

Step 1 - Create Files for Sending

The business application creates one or more **application files** containing the raw data to be sent to various trading partners. The format of the application data is typically pre-determined by the particular business application being used. Often, the business application may already exist before Digital DEC/EDI is deployed.

While it is possible for an application file to contain the data for a single document destined for a single trading partner, it is more normal for an application file to contain data for multiple documents destined for multiple trading partners. An application file often contains the net result of a particular “run” of a business application.

Step 2 - Post the Files to Digital DEC/EDI

The business application posts the application file (or a collection of application files) into Digital DEC/EDI by using the POST function of the Application Client, and specifying various parameters. The most important parameters are the **Application ID** of the business application, and the name of the **Mapping Table**.

The Application ID is the name by which Digital DEC/EDI recognizes business applications. All business applications that require use of Digital DEC/EDI must be registered with the Server (as part of the process of system configuration) along with attributes of that application, for example, on which computer nodes that application may exist.

The Mapping Table is the collection of instructions that tell the Mapping Services how to map data between an application file and resultant EDI documents.

The most simple Digital DEC/EDI system may have just one Application ID defined, and may use just one Mapping Table. More substantial installations may use many different business applications, each using one or more separate Mapping Tables. Compiled versions of Mapping Tables must be copied into the directory that forms the **Mapping Table Repository** before they can be used by the Mapping Services to process EDI data.

Step 3 - Map the Files into EDI Documents

The Mapping Services (located on the server) receive the application file posted from the Application Client and uses the specified Mapping Table to create one or more separate EDI documents. If a single application file contains the data for 100 different EDI documents, then the Mapper creates 100 separate EDI documents. These may be tracked individually through the Digital DEC/EDI system. See *Tracking Documents Through Digital DEC/EDI* on page 3-16. Each document is stored in a separate file in **internal** file format. That is, the form in which the EDI data is stored within the file is that defined internally by Digital DEC/EDI. In this form, a **data label** is assigned automatically to each item of data within the document to identify the item's meaning and position within the document.

Once an application file has been mapped into separate EDI documents in their internal file format, the processing of the application file is regarded as successful, and the Application Client returns an appropriate status to the business application.

The subsequent stages of processing happen automatically without interaction from the business application.

Step 4 - Convert the Files

The Digital DEC/EDI Translation Services contain two distinct components that deal with the processing of outbound data: the **Converter** and the **Transmission File Builder** (TFB).

The Converter takes the internal file for each EDI document and creates a corresponding **external** file for the given document. For example, where the internal file contains data for an EDIFACT invoice, the Converter produces an external file that contains the EDIFACT invoice. It does this by using the EDI Document Table definitions that form part of the Digital DEC/EDI system configuration. If the EDI Document Tables define any variations to the invoice definition as supplied by EDIFACT, then the Converter takes account of these variations.

Once the Converter has produced an external file from the corresponding internal file, the EDI document whose data is contained in these files is said to have been **converted**.

Step 5 - Build Transmission Files

The TFB is responsible for combining or **building** the EDI data from separate external files into **transmission files**, each comprising one or more interchanges of EDI data. It does this by using a collection of rules that govern when “enough” data has been built into one transmission file, and when a new one should be started. A transmission file may contain only data destined for a *single trading partner* - since it is sent to that trading partner intact).

The TFB normally builds transmission files at user-specified **build intervals**, that are normally in the range of minutes to hours, depending on the volume of EDI data being processed. It is possible to override this scheduling for individual EDI documents if they are configured or specified as **immediate** (high) priority. By default EDI documents have **normal** priority.

As part of the configuration of Digital DEC/EDI, it is possible to specify whether the TFB should apply **batching** rules. As the Mapping Services process a given application file, it may be configured to mark each resultant EDI document as belonging to a particular batch (so long as the files are also

destined for the same trading partner). Where the TFB detects EDI documents belonging to a batch, it builds the external files that form that batch into the same transmission file, in the same order as they were originally provided to the Digital DEC/EDI system. The TFB only begins the building of such a transmission file when all the documents that form the batch are available to it.

The Trading Partner Tables form a major part of the Digital DEC/EDI system configuration. Each trading partner is identified by a **Partner ID** label that is unique within a given Digital DEC/EDI system. For each trading partner is stored data about individual trading partner **agreements**. Each agreement defines the exchange of specific types of document between that trading partner and a named business application (specified by its Application ID). The definition specifies a collection of **enveloping** parameters that the Translation Services components (in particular the TFB) use to format the data in each interchange within the transmission file correctly. These enveloping parameters need to be unique, both within Digital DEC/EDI and within the trading partner's EDI system, because they are the means whereby both EDI systems can recognize who sent or received the data, and what type of data it is. The determination of trading partner agreement information is one of the key activities in setting up an EDI trading relationship.

Step 6 - Route Files to the Correct Communications Link

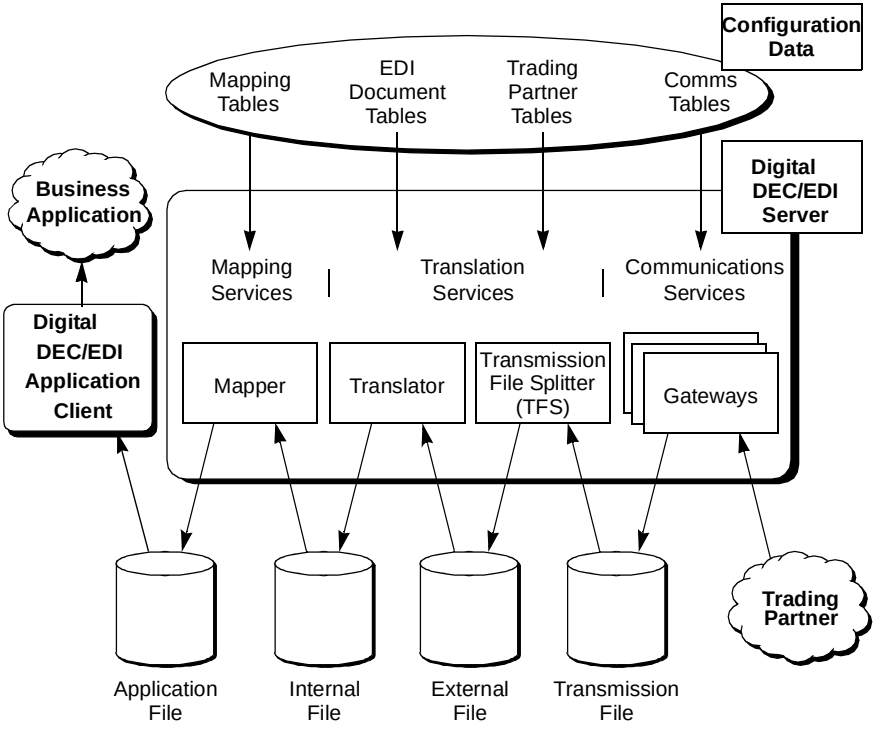
The Communications Services consist of a number of separate communications **gateways** responsible for the transfer of transmission files to trading partners. Each gateway supports one communications protocol (See *Communications Services* on page 2-11). The main element of the configuration of the Communications Services involves the specification of the parameters that describe the communications links to each destination: a destination is either a trading partner or Value Added Network (VAN) Service that takes responsibility for routing to separate trading partners. Each communications link is identified to Digital DEC/EDI by using a unique **connection ID**. The Trading Partner Tables specify which connection ID is to be used to transfer transmission files to a particular trading partner.

Outbound transmission files generated by the TFB and are processed in sequence by the appropriate communications gateway.

Inbound Data Flow

The normal flow of inbound data through the system mirrors the outbound flow in most details. The inbound data flow is illustrated in Figure 3-2 *Inbound Data Flow*.

Figure 3-2 Inbound Data Flow



The following sections outline the steps involved.

Step 1 - Digital DEC/EDI Receives Inbound Transmission File

The Communications Services receive transmission files into the Digital DEC/EDI system. Individual transmission files are passed to the Translation Services.

Step 2 - Inbound File is Processed

The Digital DEC/EDI Translation Services contain two distinct components that deal with the processing of inbound data: the **Translator** and the

Transmission File Splitter (TFS). These mirror their counterparts that deal with outbound processing: the Converter and the TFB.

The TFS checks the EDI enveloping information within the transmission file to see that it matches with a trading partner agreement that the system was expecting. Any data for which there is no trading partner agreement is failed.

Assuming there is a matching trading partner agreement, the TFS splits the transmission file into individual EDI documents in external format files.

Step 3 - Translator Produces Internal EDI File

The Translator takes the external file for each EDI document and produces a corresponding internal file. At this point the EDI document is **available** to be fetched by a business application.

Step 4 - Business Application Fetches File

A business application issues a request through the Application Client to fetch data for it to process. As with outbound processing, the business application may supply a variety of parameters to control the detail of data to be fetched, but the main parameters are:

- The name of the application (the Application ID)
- The name of the application file to be created for the inbound data
- The name of the Mapping Table the mapper should use

Step 5 - File Mapped to External Format

The Mapper maps data from the internal file for each relevant EDI document, creating the application file in the process. The specified Mapping Table controls the detail of this mapping procedure. Once the application file has been created successfully, the operation to fetch data is deemed complete.

Alternative Document Flows in Digital DEC/EDI

There are other ways that documents can be routed through Digital DEC/EDI. For example, you can route documents from one application to another within your organization -- without using Translation or Communications Services. This is termed **Application to Application**

Routing, and is described in *See Application-to-Application Routing* on page 3-8.

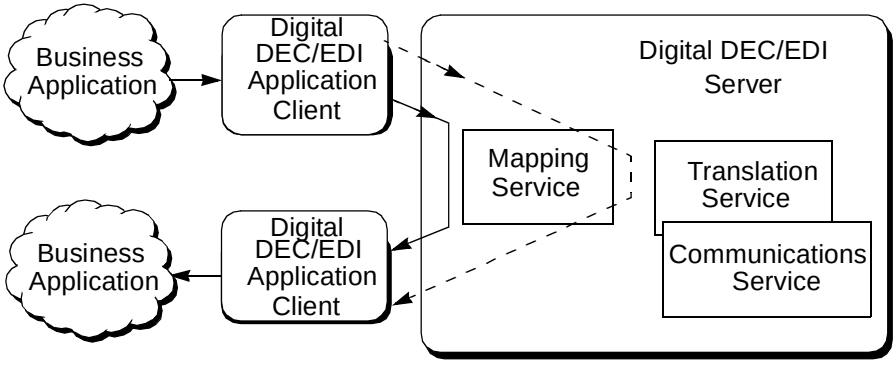
As variants of the normal data flow described in *See Digital DEC/EDI Data Flow* on page 3-1, it is possible to bypass one or other of the Mapping and Translation Services. This is termed **Bypass Routing**, and is described in *See Bypass Routing* on page 3-9.

These different methods for routing EDI data can be used in combination. While most data being routed through a given Digital DEC/EDI Server uses the normal routing, data for some applications may be routed differently. The particular routing applied to data depends on the EDI system configuration and parameters provided to the Application Client by the business application.

Application-to-Application Routing

You can route documents from one application to another, within the Digital DEC/EDI system: this is called Application-to-Application routing. You may use this facility as a means of exchanging documents between departments or organizations within the company. The routing is illustrated in Figure 3-3 *Application-to-Application Routing*.

Figure 3-3 Application-to-Application Routing



One option is for the application file to pass from the Application Client to the Server, and straight out to the other Application Client, without any processing at all. You might do this if both sending and receiving applications use the same data formats.

Alternatively, the application file can pass through the Mapping Services as an outgoing file, creating one or more internal format files in the process. These files may then be fetched back through the Mapping Services to create an application file. The two passes through the Mapping Services use different Mapping Tables, and this allows the sending and receiving applications to use different data formats.

Bypass Routing

You can route documents through just some of the services on the Digital DEC/EDI Server; this is called **bypass routing**.

As shown in Figure 3-4 *Translator Bypass Routing* and Figure 3-5 *Mapper Bypass Routing*, there are two forms of bypass routing:

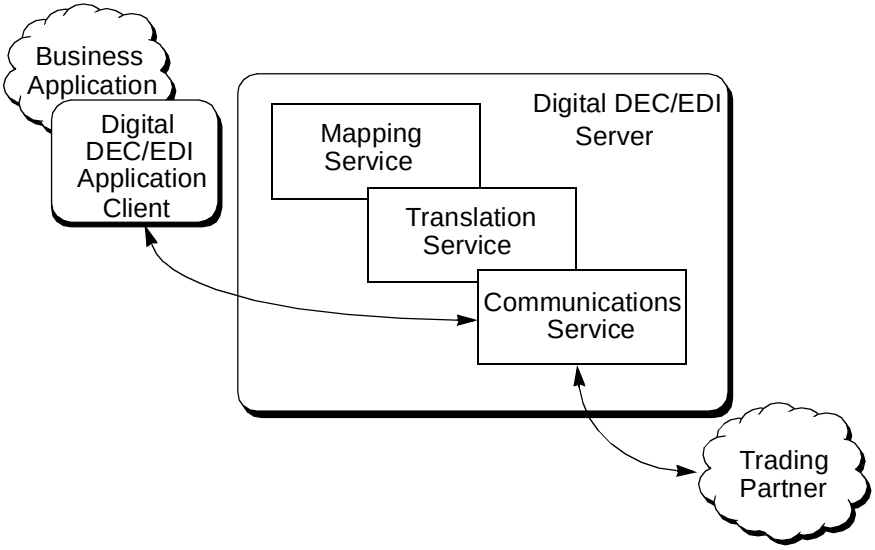
- Translator bypass routing
- Mapper bypass routing

The following sections outline their main features.

Translator Bypass Routing

Data bypasses the Mapping and Translation Services, and uses the Communications Services. Here the Server is used simply as a gateway, so that the application can exchange files with trading partners.

Figure 3-4 Translator Bypass Routing

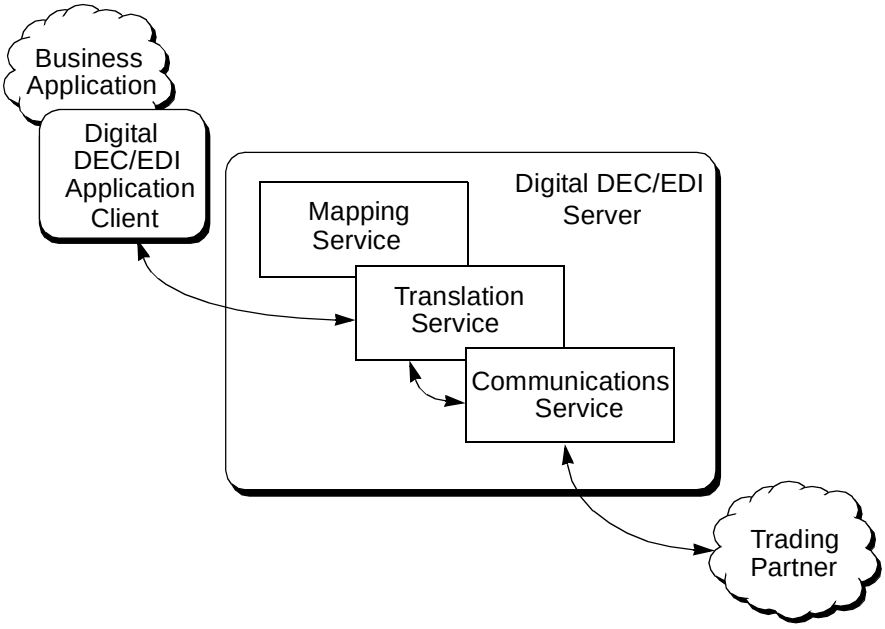


One reason for doing this is to send files of binary data, that cannot be processed into EDI format; you would have to agree this transaction with the trading partner beforehand. Alternatively you might want to send files that are already in EDI format, perhaps because they have been imported from some other EDI system.

Mapper Bypass Routing

Here, data bypasses the Mapping Services, and uses the Translation and Communications Services. You would do this if the files sent and received by the application are already in internal file format.

Figure 3-5 Mapper Bypass Routing



The main reason for doing this is to assist the migration applications from earlier versions of Digital DEC/EDI, where applications often built internal files themselves.

Note that with this form of bypass routing, the business application is not protected from any future changes to the EDI standards. For example, if in future different items of data are required by a particular version of an EDI document definition, the application will need to be modified to take account. With the normal routing of data through Digital DEC/EDI, the Mapping Services can often handle this, but with Mapper Bypass Routing, this is not possible.

Inbound Bypass Routing

With both of the bypass routing options described in *Bypass Routing* on page 3-9, the handling of outbound data is easily determined by Digital DEC/EDI when the data enters the system. The routing for inbound data is less easily determined. A transmission file received inbound may contain one or more of the following:

- A single EDI interchange
- More than one EDI interchange
- Non-EDI data (strictly, the term “non-EDI” here is used to refer to the fact that the format of the data does not conform to supported syntax definitions such as EDIFACT or X12)

The processing required by each of these differs. Accordingly, Digital DEC/EDI provides several different options that may be configured. One of the fields of configuration data for each connection allows you to define what type of data you expect to receive through the connection and, by implication, what routing is to be applied to that data. These options are relevant only to the handling of inbound data. The following options are provided:

- EDIFACT

  This option is appropriate to the normal routing of EDI data and assumes that the received transmission file contains only EDI interchanges conforming with the EDIFACT syntax. The Communications Services route the entire transmission file intact to the Translation Services.

- X12

  Like EDIFACT, this option is appropriate to the normal routing of EDI data and assumes that the received transmission file contains only EDI interchanges conforming with the X12 syntax. The Communications Services route the entire transmission file intact to the Translation Services.

- MULTIPLE

  This option is the default. It is used when the received data is expected to contain only EDI interchanges conforming with all supported EDI syntaxes. However, where the received transmission file contains more than one interchange, each interchange is split into a separate new transmission file. Each new transmission file contains a single interchange, and each is routed individually to the Translation Services.

  Any parts of the original transmission file that contain non-EDI data are placed in one or more new transmission files and marked as failed.

  While the extra processing imposed by this option may affect performance in some cases, the advantage it has over the previous options is that it allows the processing of the separate elements of the

original transmission to be monitored more closely within the Translation Services.

- BYPASS

  This option instructs the Communications Services to route the received transmission file intact, bypassing the Translation Services. This option would be used for receiving files of non-EDI data, or where the data being received was EDI data to be processed by a third-party EDI translator package.

- SPLIT

  Like the “bypass” option, the “split” option assumes that all data is to be routed bypassing the Translation Services. However if this option is selected, the Communications Services separate the received transmission file into its separate constituent parts. As with the “multiple” option, each EDI interchange and each piece of non-EDI data is placed in a separate new transmission file. These new transmission files are then individually made available to a business application.

- MIXED

  The “mixed” option makes use of an additional element of configuration data. Each agreement defined within the Trading Partner tables contains a flag to denote whether to “Use Translators On Inbound”.

  As with the “split” option, the received transmission file is split into separate transmission files, each containing either a single EDI interchange or non-EDI data. The transmission files containing non-EDI data are routed bypassing the Translation Services.

  For each transmission file containing a valid interchange of EDI data, the Trading Partner tables are searched to find an agreement whose details match the data in the transmission file. If there is a matching agreement, and if the corresponding “Use Translators On Inbound” flag is set, then that transmission file is routed to the Translation Services. Otherwise, if the flag is not set, or if there is no matching agreement, the transmission file is routed bypassing the Translation Services.

  This routing option is useful in separating inbound data, some of which is to be processed by Digital DEC/EDI and some of which is to be processed by a second EDI system. Both pieces of data can be received in the same transmission file and Digital DEC/EDI can separate the two pieces and route them accordingly.

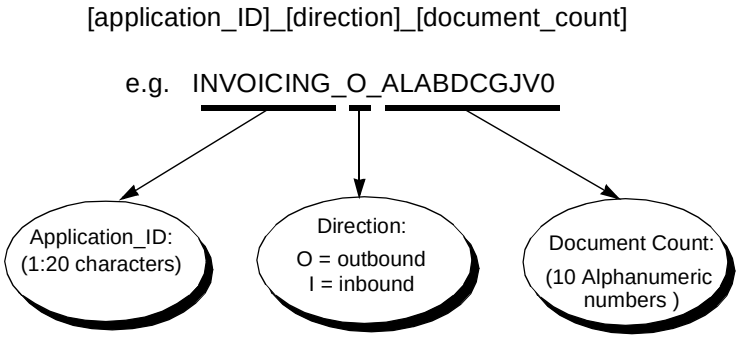
Document IDs and Transmission File Names

Each document and transmission file within the Server is allocated its own name as it is created. These names are unique within a particular Server and are the primary means for the Server to track data. Documents are referenced by their **Document ID** and transmission files by their **Transmission File Name**.

These names have a particular structure. This is shown in Figure 3-6 *Document ID Naming Convention* for documents. The Document ID is constructed from three separate pieces of information: the Application ID, the direction of the document and the document count. This document count is automatically generated by the Server using the following elements:

- cluster-member-letter(1) - Valid values are A-Z of 1 charcter width.
- seconds(5) encoded to base 36 - Valid values are 0-9 and A-Z of width 5.
- nanoseconds(4) encoded to base 36 - Valid values are 0-9 and A-Z of width 4.

Figure 3-6 Document ID Naming Convention



The convention for naming transmission files is similar. In this case the name is constructed from a date-time string that is the time at which the file was created by the Server including cluster-member-letter, plus the Connection ID. Transmission file naming convention consists of the following elements:

- Year(4) - Valid values are 0-9 of width 4.
- Month(2) - Valid values are 0-9 of width 2.
- Date(2) - Valid values are 0-9 of width 2.
- HHMMSS - Valid values are 0-9 of width 6.
- Cluster-member-letter(1) - Valid values are A-Z of 1 charcter width.
- Nanoseconds(5) Encoded to base 36 - Valid values are 0-9 and A-Z of width 4.

This is shown in Figure 3-7 *Transmission File Naming Convention*.

Figure 3-7 Transmission File Naming Convention

yyyynnddhhmmss<cluster-member-letter>
<Nanoseconds>_<connection-id>.

e.g. 20011122171120ABQDCP_ACME1_001

The naming convention for transmission files applies to all transmission files created by the Server. That is all outbound transmission files and all inbound transmission files even those imported by the Import/Export gateway. These files are imported from the directory you specify when you configure each connection used by the gateway. You must guarantee the name you give the file is of the form:

```
<transmission_file_name>_connection-ID.IMPORT
```

Even if the <transmission_file_name>_connection-ID. IMPORT file already exists in the IMPORT directory, Import/Export gateway guarantees to generate unique transmission file names (format as in fig 3-7) and DUPLICATE transmission file name error can no longer occur in Digital DEC/EDI V4.0. The resulting name, ignoring the ".transmission" extension,

must not be longer than 50 characters. Connection ID names are limited to 6 characters. An example of a valid filename might be:

```
ORDER00034256_ACME1.transmission
```

You may occasionally see transmission files with slightly different names. For example, you might see the following:

```
20011122171120ABQDCP_ACME1_001
20011122171120ABQDCP_ACME1_002
20011122171120ABQDCP_ACME1_003
...
```

The “_001”, “_002” files indicate these are inbound transmission files that have been created when the Communications Services split a larger transmission file, that in this example would have been called 22FEB199517112044_ACME1, into smaller transmission files. This happens where you configure a connection that uses the MULTIPLE, SPLIT or MIXED inbound routing options (See *Inbound Bypass Routing* on page 3-11).

Resent outbound transmission files have names prefixed by a string of the form “Rnnn_”. The “R” indicates the transmission file is a resend, the “nnn” indicates the number of the resend. The first resend is numbered “000”. For example:

```
R002_22FEB199517112044_ACME1
```

This indicates the transmission file is the third resend of the specified transmission file.

Tracking Documents Through Digital DEC/EDI

There are two methods of tracking data in the system. While they both access the same data, they suit different styles of usage.

- The Application Client TRACK function
- Digital DEC/EDI Cockpit

The following sections introduce each briefly.

The Application Client TRACK function

This is intended to provide applications with information about data in the system. These applications may be live data-processing applications wishing to see what data is available to be fetched, or how previously posted data is progressing. Alternatively, these applications may be data gathering applications run “after the event” to provide management reporting information, for example, statistics on business trading in a given period.

The Application Client TRACK function enables documents to be tracked based on business data reference fields, or references generated by the Digital DEC/EDI Server system. For example, this might include a document identifier, transmission file name, supplier code or purchase order number.

The Application Client TRACK function is not ideally suited to interactive query/response use.

Digital DEC/EDI Cockpit

This is intended to be used to see data in a Digital DEC/EDI Server in real time. It is suited to general monitoring of work in a live system and also specific query/response analysis. For example, “Is that batch of data submitted last night now processed?”. It is designed for interactive use, but is easily applied to the automated production of business reports. It requires no knowledge of commands or command line interfaces in use.

There are several items of information you can use for selecting document(s) you wish to track:

- Document ID

  Digital DEC/EDI allocates to each document a unique document ID, and to each transmission file a unique filename. You can reference these when tracking a document.

- User reference

  You can assign a user reference to each document, by which you can refer to the document: for example, a purchase order number.

- Business Data References

  Within the mapping assignments for a document, you can assign any value to business data reference fields that are stored in the Digital

DEC/EDI audit trail. You may then use these references to subsequentlt track the document, or to extract specific data values from the document.

- Various selection criteria

  You can specify various selection criteria to filter out a group of documents that you wish to track: for example, documents of a particular type, or destined for a particular trading partner, or created on a particular date, or having a particular status.

Audit Trail

The data you see when tracking documents comes from the Digital DEC/EDI audit trail. This is a log of all the document processing, maintained by Digital DEC/EDI. Every document and transmission file has an entry in the **audit trail**. The information in the audit trail is essential for several reasons:

- It enables you to track the progress of documents in the system; the tracking commands read the audit trail.
- It enables you to identify documents that have failed at some point in their processing; you can ask to see the status of documents.
- It provides an audit of all documents in the system.
- It enables you to track documents using business data reference fields.
- It enables Digital DEC/EDI itself to keep track of the documents it is processing.

Auditing and Archiving

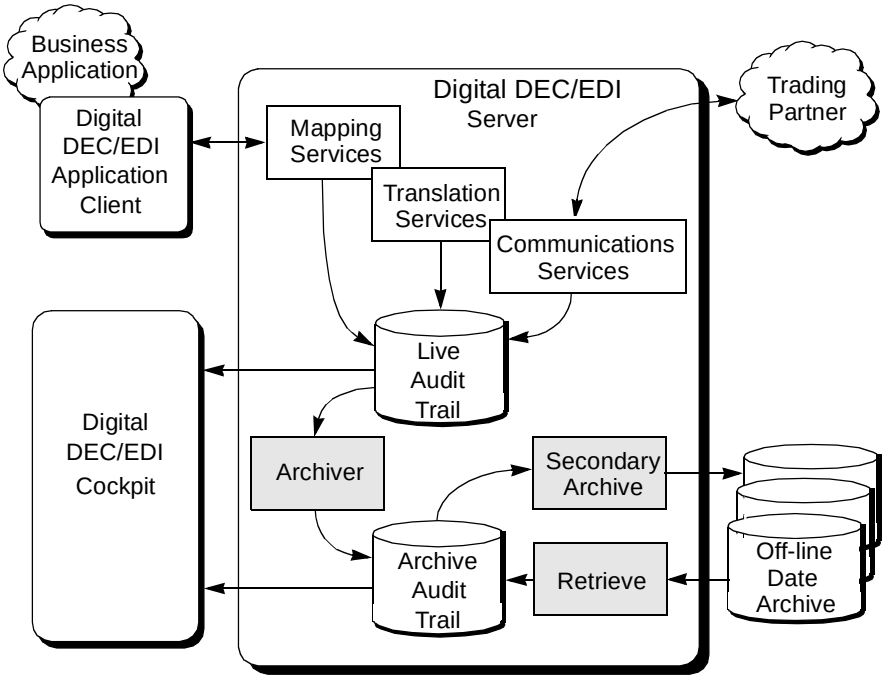
As the Digital DEC/EDI Server processes EDI data, it creates audit trail entries. That is, it keeps a record of all work being processed and that has been processed. Many different parameters are audited for each document being processed. These include:

- Its name (document ID or transmission file name)
- Its status
- The data and time it was processed
- What sort of document it was
- Which application and trading partner sent and received it
- What EDI enveloping parameters were used

In addition to this information about the document, Digital DEC/EDI also maintains copies of the data itself in each of its different formats.

All this information is created in the **live audit trail** (See Figure 3-8 *Audit Trails and Archiving*) . The size of this audit trail increases each time a new entry is created. To prevent it becoming unmanageable and slowing the system excessively, Digital DEC/EDI has an **Archiver** process. As documents and transmission files become PURGEABLE or CANCELLED, that is, no further processing is to take place, the Archiver moves them from the live audit trail to the **archive audit trail**. The archive audit trail contains “completed” EDI transactions. Both audit trails are easily viewable by using the Cockpit.

Figure 3-8 Audit Trails and Archiving



Data in the archive audit trail cannot be replaced into the live audit trail. However if a completed transaction needs to be resent, for example because a trading partner's EDI system lost a document, the Cockpit is able to take a copy of the original data from the archive audit trail and resend it. It creates a new document in the live audit trail.

The archiving process is automatic: the archiver is informed of completed data by the other components in the system. However, over time, the archive audit trail grows to a point where is must be purged of data that is no longer needed. If this does not take place, the time taken to access the archive audit trail becomes longer and longer, and may reach a level where it impacts the performance of the remainder of the system. Also, the amount of disk space required to store this information eventually reaches a point where the hardware is unable to cope.

A **secondary archive** procedure is provided to remove entries from the archive audit trail periodically. This procedure needs to be invoked through

the command line, either manually, or more normally through user-defined script files, scheduled to run periodically. The frequency with which this needs to be done depends on a variety of factors, such as the volume of data being processed, the degree to which maintaining high performance is critical, and the amount of disk space available. A **retrieve** procedure is also provided to allow the reverse operation, should there be a need to bring back for examination or to resend data removed from the system.

Digital DEC/EDI Error Log

Each of the different components of the Digital DEC/EDI Server is capable of reporting informational and error messages during operation. Informational messages include, for example, messages logged by each component during startup or shutdown. Error messages include, for example, failures in reading or writing files.

These messages are recorded in the Digital DEC/EDI Error Log. Each message logged is tagged with the name of the process that logged it, and the date and time it was logged.

You use the Cockpit to view the contents of the Error Log.

Chapter 4 Connecting Applications to Digital DEC/EDI

This chapter identifies the ways in which business applications communicate with Digital DEC/EDI.

Overview

The Digital DEC/EDI Application Client provides functions that enable files of application data to be posted to or fetched from the Digital DEC/EDI Server, or tracked within the Server. These functions may be invoked either by using script or command files, or by using embedded calls within a suitably coded application program.

The functions may be invoked from a script file, by using the Application **Client's Command Line Interface** (CLI). Alternatively, an application may contain embedded calls to the Application Client's **Application Programming Interface** (API).

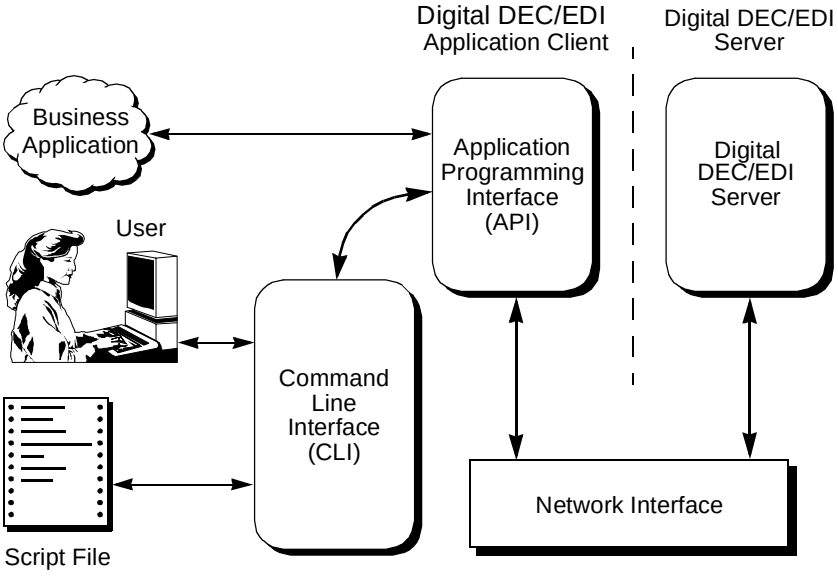
Figure 4-1 *Digital DEC/EDI Application Client Interfaces* shows the relationship between the two interfaces of the Application Client and the Digital DEC/EDI Server.

Communications between the Application Client and the Server is handled by the Digital DEC/EDI network interface. The following network interfaces are available:

- TCP/IP Only…

  This uses the TCP/IP protocol to enable the Application Client and Server to communicate.

Figure 4-1 Digital DEC/EDI Application Client Interfaces



The Application Client provides the following functions, available through both the CLI and API:

- POST

  For a business application to provide outbound application data to Digital DEC/EDI. Data can also be posted in batch mode.

- FETCH

  For a business application to fetch inbound application data from Digital DEC/EDI.

- TRACK

  For a business application to track the progress of application data within Digital DEC/EDI and to obtain information about that data, for example, unique reference numbers.

Full details of the Application Client interface, and how to use it, are contained in the Digital *DEC/EDI: Application Development book.*

Using The CLI or The API

Whether you decide to use the CLI or API to interface an application to Digital DEC/EDI depends on a variety of factors. This section highlights some of the considerations:

For the Command Line Interface:

- It is quick and easy to get started: script files are much simpler and easier to create than program code.
- Because script files are much easier to modify than program code, the CLI is better suited to initial development or production use where changes are anticipated.
- It may be the only practical solution where the application is already written, and modifications are not possible.

For the Application Programming Interface:

- It provides a much better integrated solution than using the CLI: application parameters are directly available and easier to program than with the CLI.
- The closer integration provides a faster performing solution than the CLI (although this can depend on how the interface is used).

While you can also use the CLI interactively (as opposed to using a pre-defined script file), the large number of possible parameters and values you can enter may make this approach cumbersome for other than occasional use. In particular, rather than using the tracking function interactively to monitor data in the Server, you are recommended to use the Digital DEC/EDI Cockpit.

Command Line Interface

The command name is trade. The CLI supports commands entered either from the command prompt or from a subcommand mode. The following examples show commands entered by using each of these methods:

- Entering a single command at the command prompt:

```
# trade post my-application test_file.dat \
-partner_name=the-corp \
```

```
    -tracking_reference=”ref_#1234” \
    -table_name=map_test_tbl
    # ...
```

- Using several separate commands in the subcommand mode:

```
       # trade
    CLIENTEDI> post my-application test_file.dat \
                            -partner_name=the-corp \
                            -tracking_reference=”ref_#1234” \
                            -table_name=map_test_tbl
       ...
       ...
    CLIENTEDI> fetch my-application test_file.dat \
                            -partner_name=the-corp \
                            ..
                            ..
       ...
       ...
    CLIENTEDI> exit
    #
```

These commands are normally combined with other commands in script files, as shown in the following example. This simple example posts n files into the Digital DEC/EDI Server, where n is a parameter to the script.

```
#!/bin/csh
#  P1 = Number of documents
#  ++++++++++++++++++++++++
#
set start_time=’date’
set count=1
send_one :
trade post my-application test_file.dat \
                     -partner_name=the-corp \
                     -tracking_reference=”ref_#1234” \
                     -table_name=map_test_tbl
echo “Sent document “ $count
@ count++
if ($count > $1) goto all_sent
goto send_one
all_sent:
set end_time=’date’
echo “Finished...”
echo “”
echo “Send Start time   = “ $start_time
echo “Send End time     = “ $end_time
exit 0
```

Application Programming Interface

The Application Programming Interface (API) provides C language bindings for a set of functions that can be used by business applications to interface to Digital DEC/EDI. There is a single common interface across all Application Client platforms. This means that applications written to run on one platform can easily be made to run on another.

The API provides a number of routines that an application program can use to post, fetch and track files. Note that the API routines use item lists to allow the passing of any number of optional parameters. The item lists are constructed by using tag-value pairs.

- Posting Files

  To post files, the application uses the DECEDI_POST routine to submit one or more files to the Digital DEC/EDI Server.

  The application builds up the request in a series of calls to the DECEDI_ADD_ITEM_LIST routine, and then passes the resultant item lists into the DECEDI_POST routine. When the call is completed, the application program frees the item lists by using the DECEDI_FREE_ITEM_LIST routine.

- Fetching Files

  To fetch files, the application uses the DECEDI_FETCH routine to fetch one or more files from the Digital DEC/EDI Server.

  The application builds up the request in a series of calls to the DECEDI_ADD_ITEM_LIST routine, and then passes the resultant item lists into the DECEDI_FETCH routine. When the call is completed, the application program frees the item lists by using the DECEDI_FREE_ITEM_LIST routine.

- Tracking Files

  The application uses the DECEDI_TRACK routine to track objects in the Digital DEC/EDI Server.

  The application builds up the request in a series of calls to the DECEDI_ADD_ITEM_LIST routine, and then passes the resultant item lists into the DECEDI_TRACK routine.

  The Application Client returns a list of objects that meet the selection criteria specified in the call to the DECEDI_TRACK routine. The type of information returned about each object is also specified in the call. The

returned data is passed back as a dynamic list structure that should be deallocated by using the DECEDI_FREE_TRACK_LIST routine. When the call is completed, the application program also frees the item lists by using the DECEDI_FREE_ITEM_LIST routine.

Chapter 5 Setting Up and Running Digital DEC/EDI

The setting up and running of a Digital DEC/EDI system involves a number of related tasks. These are described in more detail in other books in the Digital DEC/EDI set; see the Preface of this book for a list of titles. The following sections outline the main tasks:

Install Digital DEC/EDI

One installation guide is provided that covers installation of the Application Client and Server components, and installation of the Cockpit and CommandCenter components. The installation book also covers basic system configuration, including the network and database components, sufficent to enable verification that any installed components may communicate with one another, and are functioning correctly.

This book details things you need to consider before installing the software (for example, does your computer have enough disk space, and do you have the necessary software licenses) the installation procedure itself and any tasks you need to perform after the installation has completed. This book also describes the software products you need in addition to Digital DEC/EDI. You are strongly recommended to review this book before installing Digital DEC/EDI.

Integrate Applications with Digital DEC/EDI

The details of this activity depend on:

- Whether the application already exists or needs to be written.
- Whether you wish to use the Application Client CLI or API (See*Using The CLI or The API* on page 4-3).
- How the data needs to be routed through the Digital DEC/EDI Server (See *Digital DEC/EDI Data Flow* on page 3-1).

In a typical deployment of Digital DEC/EDI, the application already exists and a simple script needs to be created to post and fetch application files to the Application Client. In addition, one or more Mapping Tables need to be developed for use by the Mapping Services to define the mapping of data between the application file format and the Digital DEC/EDI internal file format.

For more information, see the *Digital DEC/EDI: Application Development book*, and the on-line help library information provided with the Digital DEC/EDI CommandCenter.

Configure Digital DEC/EDI

The main stages of configuration are:

- Deciding which Translation and Communications Services you want to run on your Server.
- Creation of EDI Document Tables - to define the structure and content of individual EDI documents to be exchanged with trading partners. The majority of this information is supplied by the Message Update Service, but variations to the standard definitions need to be created manually.
- Registering Business Applications - business applications that expect to use the Server must be registered before than can be used. You also register the names of nodes on which they can run.
- Creation of Communications Data - to define the configuration details of each communications link you expect to use.
- Creation of Trading Partner Profiles - to define who your trading partners are, what documents you plan to exchange with them, what EDI enveloping information you have agreed to use, and what communications links you use.

All the above data is created and maintained by using the various editors in the CommandCenter.

Detailed information on configuration is provided in the *Digital DEC/EDI: User's Guide*, and the on-line help library information provided with the Digital DEC/EDI CommandCenter.

Run Digital DEC/EDI Daily

This involves starting up and shutting down the Digital DEC/EDI system, tracking the flow of documents, identifying failed documents and either cancelling or reprocessing them, identifying problems and finding solutions.

Monitor with the Cockpit

To monitor documents and transmission files within the Digital DEC/EDI Server, use the Digital DEC/EDI Cockpit. By using the Cockpit you can look for failed documents and keep a check that the general flow of data through the system is progressing well. Once any failed documents have been identified, the reasons for the failures need to be determined and addressed before these documents are reprocessed. For more information on using the Digital DEC/EDI Cockpit, refer to the on-line help library shipped with the product.

Maintain Regularly

Periodically, and depending on how heavily used the system is, and how much spare capacity there is on your system, you need to perform one or more maintenance activities (such as archiving EDI transactions that you no longer need to keep on-line). In general, the impact of not performing such maintenance is that the system gets slower to the point where it may become unusable. Additionally the system may run out of disk space. This latter failure would result in the system halting.

See the *Digital DEC/EDI: User's Guide* for details of what maintenance activities you should plan for and also on how to approach problem solving when things go wrong. The Digital DEC/EDI Cockpit contains the *Digital DEC/EDI: Error Messages Help Library.* This contains specific advice on why certain errors appeared and what action needs to be taken.

A Note on Testing

Depending on the scope and complexity of the EDI system you want to create, you may follow each of the above steps in a pilot or test phase, either within your own organization, or with the help of a cooperative trading

partner before committing resources to full production operation. If resources permit, it is a sensible precaution to establish separate test and production environments on different computers. This allows you to test significant application modifications or configuration changes, or upgrades to pre-requisite software products, or even Digital DEC/EDI itself before committing them to production.

Chapter 6 New Features in DEC/EDI V4.0

This document presents you with a synopsis of changes made to the older version of the DEC/EDI product. The changes have also been described, in brief, in the right hand column of the table. These changes have been described in greater detail in the DEC/EDI Version V4.0 Release Notes.

In addition to multiple instances of Archive Server, Converters and Translators, the new version supports the multiple instances of Builder, Schedule and Splitter processes on a single node. The maximum number of instances to run is controlled by logicals on OVMS platform. However, Data Server, Port Server and Gateways will continue to run as single threaded application on a single node. The multiple instances of processes contribute towards improving the effective thoughput of the system.

New Feature	OVMS /Tru64	Remarks
Application Services		
Application File Server	OVMS/ Tru64	This process has been removed in the new improved version in order to reduce bottlenecks during the processing of documents
CSF	OVMS	A new AVAIL_TABLE has been added to the database. When the client initiates a “fetch”, CSF process reads the matching entries from this table

New Feature	OVMS /Tru64	Remarks
Translation Services		
Transmission File Builder (TFB)	OVMS/ Tru64	Each TFB process now writes directly to the gateway process after building the transmission file. This reduces the load on the Communications Controller. In case multiple TFBs are started, the first TFB additionally keeps track of build intervals and processes low priority documents
Transmission File Splitter (TFS)	OVMS/ Tru64	Each TFS process now separates the interchanges into individual documents in parallel
Translator	OVMS	Translators write information on successfully processed documents to the new AVAIL_TABLE. In addition, when acknowledgements are not requested for, any translator can process any of the seperated documents
Communication Services		
Node-specific connections	OVMS/ Tru64	The new version has a node-name table as an enhanced feature. This allows for configuring connections to run on specific nodes only. For more information refer to section 22-39 of OpenVMS User Support Manual–Vol 1.
Communication Controller (CC)	OVMS/ Tru64	Each CC now processes inbound connections only with the exception of first CC, which also works as a connection scheduler

New Feature	OVMS /Tru64	Remarks
General		
Decedi_startup	OVMS	The startup script starts the DEC/EDI server in full-mode. A new node-specific startup file has been added to contain configuration information of each node, thereby providing a high degree of flexibility and customisation on the node
Transmission Filename & Document ID	OVMS/ Tru64	The naming conventions used in the new version have allowed for more accurate classification of documents. It helps in primary key validation and also supports clustering mechanism. For more information refer to section 3-14 in Introduction book.
Port-id changes	OVMS/ Tru64	DECEDI_SERVERS.DAT file has now included port-id of processes against each application and server name combination. This helps in connecting to different servers from a single client node without changing the it's configuration
DHCP Support	OVMS/ Tru64	PC client running CommandCenter and Cockpit can be configured with DHCP addresses in the new version
Mailbox to Intra Cluster Connect calls conversion	OVMS	This feature improves the inter-process communication mechanism
Node specific startup file	OVMS	DEC/EDI uses a new node specific startup file suffixed with node_name. This file contains the environment logicals that govern the processing on this node. The startup file automatically determines the node-name on which it is being started in the cluster environment and copies a template file to node-specific file. For more info refer to section 2-10 in OpenVMS User Support Manual Vol-1

New Feature	OVMS /Tru64	Remarks
Size of data label increased	OVMS/ Tru64	The maximum size of data label for outgoing document is increased from 512 bytes to 32767 bytes
In-Memory queue Viewer	Tru64	DEC/EDI V4.0 on Tru64 platform maintains the status of all documents in memory. Consequently, Cockpit has been enhanced to display the various memory queue contents attached to converter, builder and translators
New Clients		
RedHat Linux	RedHat Linux	DEC/EDI Client is now available on RedHat Linux platform
Windows NT/2000 Client	WNT Client	DEC/EDI Client is now available on Windows NT/2000 platform
Database- related		
Intermediate status	OVMS/ Tru64	In the new improved version, intermediate database status has been removed, thereby reducing the databases accesses
Filebridge database	OVMS	In the new improved version, filebridge database has been merged into DEC/EDI's audit database
"YEAR_S" Field	OVMS/ Tru64	This field has been to archive database in order to improve the search accuracy of the documents spanning over multiple years
Node Name table	OVMS	DECEDI$NODE_CONFIGURATION.DAT. A new node name table that holds the registrations of each connection on the specified node(s)
Node Name table	Tru64	(scfnde) A new node name table that holds the registrations of each connection on the specified node(s)

New Feature	OVMS /Tru64	Remarks
AVAIL_TABLE	OVMS	A new table has been added to Oracle RDB database to store all the available entries in the TLF_TABLE
REPSEP_C	Tru64	TPA, TLF, and CLF (Audit & Archive tables). To store the repetition separator
REPSEP_C	OVMS	DECEDI$AGREEMENTS_TABLE DECEDI$ARCHIVE_DATABASE.SQL DECEDI$AUDIT_DATABASE.SQL DECEDI$$ARCHIVE_DB_CREATE.SC DECEDI$$DB_CREATE.SC
ELEREP_L	Tru64	STF, PST, and TST tables. To store the element repeat count for X12 004030 and above support.
ELEREP_L	OVMS	DECEDI$SEGMENT_TABLE DECEDI$P_SEGMENT_TABLE DECEDI$TP_SEGMENT_TABLE
EDIXML_C	Tru64	TPD, TLF, and CLF (Audit & Archive tables). This field indicates whether to generate EDI/XML data
EDIXML_C	OVMS	DECEDI$AGREEMENTS_DETAIL_ TABLE.FDL DECEDI$ARCHIVE_DATABASE.SQL DECEDI$AUDIT_DATABASE.SQL DECEDI$$ARCHIVE_DB_CREATE.SC DECEDI$$DB_CREATE.SC
XMLTAGNAME_C	Tru64	TPD table. This field indicates whether the Datalabel/Description is used as TagName
XMLTAGNAME_C	OVMS	DECEDI$AGREEMENTS_DETAIL_ TABLE.FDL
XMLUNDERSCORE_C	Tru64	This field indicates whether the underscore can be used in the TagName. For ex: LineItem or Line_Item
XMLUNDERSCORE_C	OVMS	DECEDI$AGREEMENTS_DETAIL_ TABLE.FDL

New Feature	OVMS /Tru64	Remarks
XMLATTRIBUTE_C	Tru64	SEF, PSE table. This field is used to differentiate whether the element is attribute or not. For ex: <Purchase order="12345">
XMLATTRIBUTE_C	OVMS	DECEDI$SUB_ELE_TABLE.FDL DECEDI$P_SUB_ELE_TABLE.FDL
SEGDESC_S	Tru64	SDF, PSD table. This field contains the segment description
SEGDESC_S	OVMS	DECEDI$SEG_DESC.FDL DECEDI$P_SEG_DESC.FDL
SUBELDES_S	Tru64	SEF, PSE table. This field contains the element description
SUBELDES_S	OVMS	DECEDI$SUB_ELE_TABLE.FDL DECEDI$P_SUB_ELE_TABLE.FDL
COMPDESC_S	Tru64	CED, PCD table. This field contains the composite element description
COMPDESC_S	OVMS	No tables are there to store the composite values
DOCDES_S	Tru64	DDF, PDF table. This field contains the document name description
DOCDES_S	OVMS	DECEDI$DOC_DESC_TABLE.DAT DECEDI$P_DOC_DESC_TABLE.DAT

Glossary

This glossary explains some of the terminology that you may come across when using Digital DEC/EDI. The entries are in alphabetic order. References from one entry to another are in italic.

Acknowledgement

Reply sent in response to a received message to indicate to the originator that the message has been received and can be satisfactorily be processed.

Agreement

See *Trading Partner Agreement.*

ANA

Article Number Association. An Independent organization developing EDI standards for business in the United Kingdom.

ANSI

American National Standards Institute.

API

See *Application Programming Interface*.

Application

See *Business Application*.

Application Client

Business applications interface to the Digital DEC/EDI system by using the Application Client. The Application Client provides the ability for business applications to post file-based EDI data into the system or to fetch data from the system. Business applications can also track the progress of data in the system.

Application File

The files of EDI data that business applications post to or fetch from Digital DEC/ EDI, via the Application Client, are called Application Files.

Application ID

Each business application that needs to use the Digital DEC/EDI system must first be registered with the system, by using a unique name for that application. This name is the Application ID.

Application Programming Interface

The Application Client presents business applications with functions to POST, FETCH and TRACK EDI data. It provides these functions through both command line and programming interfaces, the latter being referred to as the Application Programming Interface (API). The API allows program calls to the functions provided by the Application Client to be embedded directly into business applications.

Application-to-Application Routing

EDI data posted through an application client is normally routed through Mapping, Translation and Communications Services in sequence, and from there, to a trading partner. Inbound data is normally routed through the reverse path.

Application-to-Application routing offers a different approach. It allows data from a business application to be posted into the Digital DEC/EDI system from the Application Client to the Server, and then immediately fetched by a second business application through an Application Client. Note that this routing option uses the Server to provide auditable evidence of the transactions: full audit trail entries are created for all data routed this way.

Archive Audit Trail

See *Archiving*.

Archiving

As EDI data is processed by the Digital DEC/EDI system, an audit trail is maintained to provide a record of information about the data and a copy of the data itself. While the data is being processed by Digital DEC/EDI, this information resides in the Live Audit Trail. Once processing of a piece of data is completed, its audit trail entry is removed and placed in the Archive Audit Trail. This process is referred to as Archiving, and is performed by the Archiver process.

Areas

The data for a document can typically be divided up into “areas” based on the nature of the data. The standards bodies thus divide their documents into areas thus:

- A header area, which contains information pertaining to the whole document, e.g. name and address of sender, and references.
- A Details area, which contains the line items of the document, e.g. the list of items being ordered in a purchase order document.
- A VAT area, which contains details of the VAT rates used in the document (Tradacoms only).
- A Summary area, which contains totals for the line items, e.g. total amount due in an invoice.

Not all documents use all of these areas, however the order of the areas in a document must be as listed above.

Note that “area” is a generic term for this concept in Digital DEC/EDI. In EDIFACT/Odette the term “area” is used, however in X12/TDCC the term “section” is used, in Tradacoms “message” is used.

ASCII

American Standard Code for Information Interchange.

Audit Trail

As EDI data passes through the Digital DEC/EDI system, information about the data and copies of the data itself are maintained in an audit trail. This provides auditable evidence of the processing of the data and, if needed, allows the data to be resent. See also *Archiving*.

Batch

As the Mapper is processing Application File data, it may be configured to mark consecutive EDI documents as being from the same batch. That is, the data comes from a single application file and is destined for the same trading partner. When the Transmission File Builder (TFB) receives documents belonging to a batch, it builds them all into the same transmission file in the same order they were supplied to the system. The TFB does not start to build the transmission file unless all documents belonging to the batch are available to it. This guarantees both the order and the presence of all the documents belonging to a batch within a particular transmission file.

Business Application

A Business Application is a computer program that uses the Application Client to post, fetch or track EDI data.

Bypass Routing

The normal means of routing outbound EDI data through the system is Application Client, Mapping, Translation, and Communications Services respectively. Inbound data is routed through these Services in the reverse order. Bypass Routing refers to the means of routing data to avoid one or both of the Mapping and Translation Services. See also *Translator Bypass Routing and Mapper Bypass Routing*.

CLEO

CLEO is a registered trademark name for CLEO Communications. When this name is used, it refers to CLEO Communications. CLEO Communications produces the 3780Plus product.

CLI

See *Command Line Interface*.

Cockpit

The Cockpit is one of the components of the Digital DEC/EDI system. It forms a graphical user interface that runs on a PC under MS-Windows. It allows the user to view, either in summary or detail, the information in the live or archive audit trails, and the Error Log.

CommandCenter

The CommandCenter is one of the components of the Digital DEC/EDI system. It forms a graphical user interface that runs on a PC under MS-Windows. The CommandCenter allows the user to create and maintain the necessary configuration data that the Server uses.

Command Line Interface

The Application Client presents business applications with functions to POST, FETCH and TRACK EDI data. It provides these functions through both command line and programming interfaces, the former being referred to as the Command Line Interface (CLI). The CLI allows simple business applications to be created by using easily created script or command files.

Communications Services

The Communications Services form one of the three groups of services provided by the Digital DEC/EDI Server. They provide support for sending and receiving EDI data, either directly to trading partners or through third party VAN services by implementing a range of different communications protocols. See also *Mapping Services and Translation Services*.

Connection ID

Each communications link with a trading partner or VAN service that Digital DEC/ EDI uses to transfer EDI data has a unique name. This name is the Connection ID.

Converter

The Converter is a component of the Translation Services responsible for the conversion of outbound EDI data from internal file format to external file format. That is, the component is responsible for producing EDI documents in a form that complies with both the relevant EDI syntax requirements and EDI document definitions.

Digital DEC/EDI

The name given to the collection of products that form Digital's EDI system. A Digital DEC/EDI system is created by the deployment and configuration of one of more copies of the Digital DEC/EDI system components: Application Client, Server, Cockpit and CommandCenter.

Data Label

A label used to tag a particular piece of data used to indicate its meaning and position. The internal file format uses data labels extensively to tag each item of data.

Dictionary

This is the collective term used to refer to the hierarchy of individual pieces from which EDI document definitions are created. A dictionary for a particular version of an EDI standard contains definitions for segments and elements.

Document

Digital DEC/EDI uses the term Document to refer to an individual EDI document, such as one invoice or one purchase order. Each document within

the Digital DEC/EDI system has a unique label: its Document ID. The Document ID can be use as a parameter for tracking documents in the system.

Document ID

See *Document*.

Document Type

A definition of a specified class of document, for example, an invoice.

EDI

See *Electronic Data Interchange*.

EDIFACT

EDIFACT is an acronym for Electronic Data Interchange for Administration, Commerce and Transport. EDIFACT is an EDI Standards body responsible for the definition of EDI document definitions and dictionaries based on EDIFACT syntax.

Electronic Data Interchange

Electronic Data Interchange is the electronic transfer of structured business data between business applications, using agreed communication systems.

Element

An item of data from an EDI document, such as a part of a trading partner's address. Elements may have a simple structure or may have a composite structure, that is they may be thought of as containing two or more individual items of data. A collection of related elements form a segment.

Error Listing

Lists the errors that occurred when the translation services failed to convert or translate a document. This can be viewed from the Cockpit.

Error Log

Each of the different components of the Digital DEC/EDI Server is capable of reporting information and error messages during operation. These messages are logged in the Error Log file. A message is tagged with the name of the process that logged it and the date and time it was logged. You use the Cockpit to view the Error Log.

External File Format

Refers to the format of information in a file that contains a single EDI document formatted to conform with the appropriate EDI standard document definition, for example an EDIFACT purchase order.

Functional Acknowledgment

With X12/TDCC when sending documents you can request that a Functional Acknowledgment document is sent back by the receiver of your documents to confirm that they have been received.

Functional Group

See *Interchange*.

Gateway

The Communication Services contain a number of gateways. Each gateway provides support for one of the communications protocols that Digital DEC/EDI supports. Digital DEC/EDI provides the following gateways: *Import/Export, OFTP, Pedi, 3780 and SMTP/MIME*.

Import/Export

The Import/Export gateway allows transmission files to be exported to disk or imported from disk. It is particularly use on test systems where live communications links (or trading partners) do not exist. It is also useful where a communications protocol is to be used that is not directly supported by Digital DEC/EDI.

Inbound

Inbound data is application data fetched by a business application from the Application Client. Assuming *normal routing* such data would have been received from a trading partner and routed through the Communications Services, Translation Services and Mapping Services (in that order) before being fetched.

Interchange

An aggregate of one or more EDI documents bound together with the same enveloping details. Where documents are of a different type, for example, invoices and purchase orders, they are contained within separate functional groups within the interchange. All documents within an interchange are covered by the same trading partner agreement.

Internal File Format

Refers to a format for information in a file that contains the data from a single EDI document. In this format a data label is assigned to each item of data within the document to identify the item's meaning and position within the document. See also *External Format File.*

Live Audit Trail

As EDI data passes through the Digital DEC/EDI system, information about the data and copies of the data itself are maintained in the Live Audit Trail. Once the processing of a piece of data is completed it's entries in the Live Audit Trail are removed and copied to the Archive Audit Trail. The Live Audit Trail contains entries for EDI data currently being processed, EDI data that has failed processing and that is awaiting manual intervention, and a small number of completed entries that are being, or are about to be, removed to the Archive Audit Trail.

See also *Archiving and Archive Audit Trail*.

MUS

See *Message Updates Services*.

Mapper

See *Mapping Services*.

Mapper Bypass Routing

Outbound data posted through the Application Client is routed direct to the Translation Services and then to the Communications Services, thus bypassing the Mapping Services. Any files routed by using this bypass option must be in internal file format.

Inbound data is routed from the Communications Services and then to the Translation Services. The Application Client can then fetch this data that is provided in internal file format.

See also *Translator Bypass Routing*.

Mapping Services

The Mapping Services form the link between the unconstrained format of the application file and the more rigorously defined format of documents, as

specified by the EDI Standards bodies. The main component of the Mapping Services is the Mapper.

In operation, the Mapper uses a pre-defined Mapping Table to determine how data in the two different formats is mapped on a piece-by-piece basis.

Mapping Table

The Mapper uses a pre-defined Mapping Table to determine the mapping of individual pieces of EDI data between the application file and EDI documents.

A Mapping Table contains three main parts: a definition of the format of a particular application file type, the definition of the EDI document, and detailed mapping instructions for creating the one from the other. Different mapping tables are required to map the same data in each direction.

Mapping Table Repository

The Mapping Table Repository is the name of the directory location where the Mapper accesses compiled Mapping Tables (.FBO files) at runtime.

Message Update Services

These are a collection of tables defining the EDI documents specified by EDI standards bodies. The tables consist of the definitions of documents in terms of their constituent parts, and the hierarchical definitions of those parts. Different versions of these tables are provided to conform to the different versions of such definitions as dictated by the EDI standards bodies.

MIME

Multipurpose Internet Mail Extensions (MIME) is a mechanism for specifying and describing the format of Internet message bodies. It defines the format of message bodies to allow multi-part textual and non-textual message bodies to be represented and exchanged without loss of information. In particular, it provides facilities to include multiple objects in a single message, to represent body text in character sets other than US-ASCII, to represent formatted multi-font text messages, to represent non-textual material such as images and audio fragments, and generally to facilitate the representation of new types of Internet mail such as EDI. It is defined in RFC 1521.

Normal Routing

Routing within Digital DEC/EDI refers to flow of data between the three main services. Normal routing of outbound data posted through the Application Client is processing by the Mapping, Translation and Communications Services in that order. For inbound data, processing is in the reverse order. See also *Bypass Routing and Application-to-Application Routing*.

ODETTE

ODETTE is an acronym for Organization for Data Exchange by Tele-Transmission in Europe. ODETTE is an EDI Standards body defining EDI Document definitions based on the EDIFACT syntax. See also *ODETTE File Transfer Protocol*.

ODETTE File Transfer Protocol

ODETTE File Transfer Protocol (OFTP) refers to an X.25-based communications protocol supported by Digital DEC/EDI for the fast, reliable transfer of EDI data between trading partners. OFTP was originally used in the European motor industry, but is increasingly being deployed worldwide in other industries.

OFTP

See *ODETTE File Transfer Protocol.*

Outbound

Outbound data is application data posted by a business application through the Application Client. Assuming normal routing such data would be routed through the Mapping Services, Translation Services and Communications Services, in that order, before being sent to a trading partner.

P0

See *X.400.*

P2

See *X.400*.

Partner ID

Each trading partner with whom a Digital DEC/EDI system may exchange EDI data needs to be given a unique name. This name is the Partner ID.

Pedi

See *X.400*.

Retrieve

See *Secondary Archiving.*

Secondary Archiving

Secondary Archiving is the term that refers to the removal of data from the Archive Audit Trail to an off-line storage location. This data would not normally be accessed other than for business auditing purposes, or in extreme cases, to allow a document or transmission file to be resent. Once data is removed to an off-line storage location by the process of Secondary Archiving, it can be brought back by using the Retrieve procedure.

See *Archiving*.

Segment

An item of data from an EDI document. A segment contains the data from several related elements (for example parts of a trading partner's address).

Server

The Digital DEC/EDI Server forms the heart of the Digital DEC/EDI system providing end-to-end EDI services to authorized business applications. See *Mapping Services, Translation Services and Communications Services*.

SMTP

Simple Mail Transfer Protocol.

TDCC

See *X12*.

TFB

See *Transmission File Builder.*

TFS

See *Transmission File Splitter.*

TRADACOMS

Tradacoms is a set of EDI message standards and dictionaries controlled by the Article Number Association (ANA). They are used predominately in the UK.

Trading Partner

The name used to refer to the companies or organizations engaged in trading using EDI.

Trading Partner Agreement

Two or more trading partners create a Trading Partner Agreement prior to trading with EDI to define, what they are to exchange and how they are to exchange it. What they are to exchange is defined in terms of the agreed structure of selected EDI documents in addition to enveloping data. Enveloping data contains the necessary EDI addressing information so both sending and receiving EDI systems can identify for any one document the particular Trading Partner Agreement that applies to that document.

Translator Bypass Routing

Outbound data posted through the Application Client is routed direct to Communications Services, thus bypassing the Mapping and Translation Services. Any files routed by using this bypass option must be in transmission file format.

Inbound data is routed through the Communications Services and may then be fetched by an Application Client. The data is provided in transmission file format.

This form of bypass routing can be useful for transfer of non-EDI data or to use the Communications Services of Digital DEC/EDI with the EDI Translation Services of third-party or proprietary systems.

See also *Mapper Bypass Routing*.

Translation Services

The Translation Services form one of the three groups of services provided by the Digital DEC/EDI Server. They provide the functionality to change EDI data

to and from the internal file format and the format defined by the EDI standards bodies. For further information, see also *Converter, Transmission File Builder, Translator* and *Transmission File Splitter.*

Translator

One of the sub-components of the Translation Services responsible for translating inbound files from external file format to the internal file format.

Transmission File

A transmission file contains a structured collection of one or more EDI documents (for example a batch of invoices) within a transmission envelope. A transmission file is the physical unit of information exchanged between trading partners. DEC/EDI gives each transmission file a different name that is unique within a DEC/EDI Server. This transmission file name is used for tracking particular transmission files.

Transmission File Builder

One of the sub-components of the Translation Services responsible for building outbound transmission files (containing many documents) from the individual documents made available by the Converter.

Transmission File Name

See *Transmission File*.

Transmission File Splitter

One of the sub-components of the Translation Services responsible for splitting inbound transmission files (containing many documents) into individual documents for translation by the Translator.

VAN

See *Value Added Network.*

Value Added Network

A third party agency that provides communications (and often more) services to its subscribers. An individual subscriber need only create and maintain communications links to the Value Added Network (VAN). The VAN is responsible for the connection and delivery of data to the other subscribers (some of whom are your trading partners).

View

The general name for a particular selection of parameters from the audit trail to be included in a display within the Cockpit. A simple view might contain the document ID and the date and time the document entered the system. A more complex view might show several of the EDI enveloping parameters relating to each document.

X12

X12 is a set of EDI message standards and dictionaries controlled by the American National Standards Institute (ANSI). They are mainly used in the USA.

X.400

A store-and-forward communications protocol developed primarily for reliable transfer of interpersonal mail and also frequently used for the exchange of EDI data. The protocol defines several different communications enveloping options that may be deployed: P0, P2 and Pedi.

P0 enveloping is the most basic approach where the EDI transmission file is carried as a “body part” with a minimum of addressing (enveloping) information. Its use for the transfer of EDI data is becoming less frequent.

P2 enveloping is the most common form used for EDI. It extends the P0 approach by providing additional addressing (header) information as part of the “body part” alongside the EDI transmission file.

Pedi enveloping was specifically developed for the transfer of EDI data, whereas P0 and P2 were developed for general purpose and interpersonal mail exchange where the data content was informal. Pedi significantly extends the P2 enveloping approach by requiring many different additional header fields and also allows multiple body parts to be delivered. These additional fields provide for a variety of additional services to support things such as authentication, non-repudiation, security, forwarding of responsibility.

X.435

A set of recommendations for supporting exchange of EDI data by using X.400 communications systems. They extend the X.400 enveloping model to define a new enveloping type specific to the transfer of EDI data: Pedi.

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

X