Australian Government

Medicare Australia

MEDICARE EASYCLAIM

Logic Pack Users Guide

Version 2.0

April 2008

Copyright Notice

This document contains information protected by copyright. © Commonwealth of Australia

This work is copyright. You may download, display, print and reproduce this material in unaltered form only (retaining this notice) for your personal, non-commercial use or use within your organisation. Apart from any use as permitted under the Copyright Act 1968, all other rights are reserved. Requests and enquiries concerning reproduction and rights should be addressed to The Manager, Media, Marketing and Communications Branch, Medicare Australia National Office, PO Box 1001 Tuggeranong DC ACT 2901.

Contact

Medicare Australia
Locked Bag 6666
Tuggeranong DC ACT 2901
AUSTRALIA

Disclaimer

Although every care has been taken in the compilation of the material on this document, Medicare Australia does not assume or accept any responsibility for the accuracy, completeness, quality, authenticity, and currency of the information on this document.

Where provided in electronic form, Medicare Australia does not warrant that the information on the document is free of infection by computer viruses or other contamination.

Medicare Australia does not accept any responsibility or liability for any injury, direct or indirect loss, or damage, incurred by the use of or reliance on the information in this document. Medicare Australia recommends that users exercise their own skill and care with respect to its contents and make their own judgments about those matters.

Revision History

Date	Version	Description	Author
17/01/2007 to 01/03/2007	0.1 to 1.0	Initial Draft Skeleton Version to Final Beta Version to accompany Beta Release	Technical Writer, Medicare Easyclaim Team
14/03/2007	1.1	Beta review and additional information included & reviewed by Media & Marketing	Technical Writer, Medicare Easyclaim Team, Media & Marketing
22/03/2007	1.2	Final	Technical Writer
23/04/2007	1.3	Minor changes and corrections made	Technical Writer, Medicare Easyclaim Team
25/03/2008	2.0	Changes for R2	Medicare Easyclaim Team

Review History

Date	Version	Description	Reviewer
17/01/2007 to 29/01/2007	0.1 to 1.0	Review of source documents to Final Management review	Medicare Easyclaim Team Media, Communication and Government Relations Online Claims Online Claims, eBusiness & Development Division Solutions Branch Manager
14/03/2007	1.1	Beta review and additional information included & reviewed by Media & Marketing	Technical Writer, Medicare Easyclaim Team, Media & Marketing
22/03/2007	1.2	Final	Technical Writer, Medicare Easyclaim Team
23/04/2007	1.3	Minor changes and corrections made	Technical Writer, Medicare Easyclaim Team
26/03/2008	2.0	Changes for R2	Medicare Easyclaim Team

Approval History

Date	Version	Description	Approver
29/01/2007	1.0	Technical Approval Public Affairs Approval Project Manager Approval Management Approval Management Approval	Medicare Easyclaim Team Media, Communication and Government Relations & Online Claims Medicare Easyclaim Project Manager Medicare Easyclaim Project Sponsor Solutions Branch Manager
22/03/2007	1.2	Technical Approval Public Affairs Approval Project Manager Approval Management Approval Management Approval	Medicare Easyclaim Team Media, Communication and Government Relations & Online Claims Medicare Easyclaim Project Manager Medicare Easyclaim Project Sponsor Solutions Branch Manager
23/04/2007	1.3	Minor changes and corrections made	Medicare Easyclaim Team Medicare Easyclaim Project Manager
28/03/2008	2.0	Changes for R2	Medicare Easyclaim Team Medicare Easyclaim Project Manager

Table of Contents

1 Introduction 1
1.1 About this Guide 1
1.2 Target Audience 1
1.3 Assumptions 1
1.4 Related Documents 1
1.5 Support 1
1.6 Terminology 2
2 Medicare Easyclaim Logic Pack User Guide v2.0 4
2.1 Overview 4
2.1.1 Patient Claiming (PCe) 4
2.1.2 Bulk Billing (BBe) 4
2.2 What is a Logic Pack? 5
2.3 What is the Server Adaptor? 5
2.4 Changes in R2.0 5
3 Patient Claims (PCe) 7
3.1 Patient Claiming Message Flow 7
3.1.1 Description of sequence PCe Flow 7
4 Bulk Billing (BBe) 8
4.1 Bulk Billing process Flow 8
4.1.1 Description of sequence BBe Flow 8
5 Examples 9
5.1 Generate a Request 9
5.2 Validate a Request 10
5.3 Generate XML Code 10
5.4 Generate Java Object from XML 13
6 How to use the response 14
7 Return Codes 15

Table of Figures

Figure 1. Process Flow for Patient Claiming (PCe) 7
Figure 2. Process Flow for Bulk Billing (BBe) 8
Figure 3. Logic Process Flow 9

Table of Tables

Table 1. Request Type, Status Code, and XML Response 14
Table 2. Return Codes 16
Table 3. Patient Claiming (PCe) Request Data Elements 28
Table 4. Patient Claim (PCe) Cancel 29
Table 5. Patient Claim (PCe) Response 30
Table 6. BulkBilleClaimingRequestSH 35
Table 7. BulkBilleClaimingConfirmTypeRequestSH 35
Table 8. BulkBilleClaimingResponseHS 37
Table 9. Practice Location Character 47
Table 10. Check Digit 47
Table 11. TransactionId validation 49

Table of Appendices

Appendix A. Code Samples....17
A.1. Sample Patient Claim (PCe) Code....17
A.1.1. Sample General Service Patient Claim Code....17
A.1.2. Sample Specialist Service Patient Claim Code....18
A.1.3. Sample Diagnostic Imaging Service Patient Claim Code....18
A.1.4. Sample Cancel Request Patient Claim Code....19
A.2. Sample Bulk Billing (BBe) Request Code....20
A.2.1. Sample code for General Practitioner Service....20
A.2.2. Sample code for Specialist Service....21
A.2.3. Sample code for Pathology Service....22
A.2.4. Sample code for Diagnostic Imaging....23
A.2.5. Sample code for Confirm Type Request....24
Appendix B. Data Elements....25
B.1. Patient Claim (PCe) Data Elements....25
B.1.1. Patient Claim (PCe) Request Data Elements....25
B.1.2. Patient Claim (PCe) Cancel Request Data Elements....29
B.1.3. Patient Claim (PCe) Response Data Elements....29
B.1.4. Patient Claim (PCe) Cancel Response....30
B.2. Bulk Bill (BBe) Data Elements....31
B.2.1. BulkBilleClaimingRequestSH Data Elements....31
B.2.2. BulkBilleClaimingConfirmTypeRequestSH Data Elements....35
B.2.3. BulkBilleClaimingResponseHS Data Elements....36
Appendix C. Sample XML Messages....38
C.1. Patient Claiming (PCe) XML Messages....38
C.1.1. Sample PatienteClaimingRequestSH - General Practitioner Services XML Message....38
C.1.2. Sample PatienteClaimingRequestSH - Specialist Services Request XML Message....38
C.1.3. Sample PatienteClaimingRequestSH - Diagnostic Imaging Services Request XML Message....39
C.1.4. Sample PatienteClaimingRequestSH - Not for Comparison - Restrictive override code NC XML Message....39
C.1.5. Sample PatienteClaimingRequestSH - Separate Site - Restrictive override code SP XML Message....40
C.1.6. Sample PatienteClaimingRequestSH - Not Related Care Plan and Consultation - Restrictive override code NR XML Message....40
C.1.7. Sample PatienteClaimingResponseHS Response XML Message ExplanationCde (Length 3)....41
C.1.8. Sample PatienteClaimingResponseHS Response XML Message ExplanationCde (Length 4)....41
C.1.9. Sample PatienteClaimingCancelRequestSH Cancel Request XML Message....42
C.1.10. Sample PatienteClaimingCancelResponseHS Cancel Response XML Message....42
C.2. Bulk Billing (BBe) XML Messages....42
C.2.1. Sample Bulk Billing (BBe) General Practitioner Services Request XML Message....42
C.2.2. Sample Bulk Billing (BBe) Specialist Services Request XML Message....42
C.2.3. Sample Bulk Billing (BBe) Pathology Services Request XML Message....43
C.2.4. Sample Bulk Billing (BBe) Diagnostic Imaging Services Request XML Message....44
C.2.5. Sample BulkBilleClaimingConfirmTypeRequestSH Confirm Type Request XML Message44
C.2.6. Sample BulkBilleClaimingResponseHS Response XML Message....44
C.2.7. Sample BulkBilleClaimingConfirmTypeResponseHS Confirm Type Response XML Message....45
Appendix D. Provider and Card Validation Routines....46
D.1. Medicare Card Number Validation....46
D.2. Provider Number Validation....46
D.2.1. Practice Location Character (PLV)....47
D.2.2. Check Digit....47

Appendix E. Message Headers 48
E.1. Request Messages Headers 48
E.2. Response Message Headers 48
E.3. TransactionId 49
Appendix F. Medicare Easyclaim Schema 50
F.1. Medicare Easyclaim Schema 50
Appendix G. HICOnline Support Schema 56
G.1. HICOnline Support Schema 56

1 Introduction

1.1 About this Guide

This document is a guide for installing, configuring, and integrating the Medicare Easyclaim Logic Pack software into a financial institution's IT environment.

1.2 Target Audience

This document is designed to be used by technical personnel involved with software development or system administration related to the Medicare Easyclaim project. It assumes that its readers have a basic understanding of:

- XML
- the Java programming language
- Java servlets technology and the configuration of Java application servers ("servlet containers").

1.3 Assumptions

This document assumes the reader has some familiarity with the following software:

- Jetty: Servlet Container (optional)
- ANT: Build Tool
- Log4J: Logging Framework

1.4 Related Documents

Please refer to the following documents for additional information:

1. Medicare Easyclaim Technical Requirements Specification
   (This can be found on the Medicare Australia web site)
2. Medicare Easyclaim Server Adaptor Adopter Guide

1.5 Support

Please contact Medicare Australia Online Technical Support Helpdesk for any project, business, architecture, technical, or integration issues with the Medicare Easyclaim Server Adaptor:

Medicare Australia Online Technical Support Helpdesk

Email: **eclaiming@medicareaustralia.gov.au**
Phone: **1300 550 115** (option 4 then option 3)

Please also report any errors or omissions in this document to the Medicare Australia Online Technical Support Helpdesk.

1.6 Terminology

The following terms and acronyms are used within this document:

Term/Acronym	Meaning
ACSI 33	Australian Government Information and Communications Technology Security Manual.
AEDST	Australian Eastern Daylight Saving Time.
AEST	Australian Eastern Standard Time.
APCA	Australian Payments Clearing Association.
B2B	Business-to-Business.
BBB	Participant's Financial Institution ID as per the banking standard.
BBe	Bulk Bill Medicare Easyclaim – a message used in the transmitting of a bulk bill claim in the Medicare Easyclaim system.
BeC	Bulk Bill Medicare Easyclaim Confirm Type.
BH	Bank Hub.
BS	Bank System.
Bulk Billing	A practitioner who bulk bills undertakes to accept the relevant Medicare benefit as full payment for the service. The patient eligible for a benefit under the Medicare program assigns his or her right to the benefit to the practitioner. The practitioner then bills Medicare Australia instead of the patient.
CEV	Concessional Entitlement Verification.
DBS	Bulk Bill Claim.
DMZ	Demilitarized Zone – a computer host or network inserted as a "neutral zone" between an organisation's private network and the outside network. It prevents outside users from getting direct access to a server in the organisation's private network.
EFTPOS	Electronic Funds Transfer at Point Of Sale.
FTP	File Transfer Protocol – a standard internet protocol to exchange files between computers.
HCL	Healthcare Location.
HOP3	The messaging protocol used to communicate between the server adaptor and Medicare Australia.
HTTP	Hyper Text Transfer Protocol – the communications protocol on which the World Wide Web (WWW) is based. HTTP is implemented on top of the TCP/IP protocol, which provides a reliable connection-oriented transport service.
IDL	Interface Definition Language.
IPSEC	Internet Protocol Security is a framework for a set of protocols for security at the network or packet processing layer of network communication.
IRN	Individual Reference Number. A unique, identifying number for each person listed on a Medicare card. It appears to the left of each name on the Medicare card. It is also referred to as the sub numerate.
Item Number	Each medical service contained in the Medicare Benefits Schedule has been allocated a unique item number. Detailed with each item number and service description is the Medicare schedule fee and an explanatory note relating to the service.
JAR	Jar file.
JAXB	The Java XML Binding Architecture – a standard for mapping between XML Schemas and Java objects.
LCC	Licensed Collection Centre. Now known as Specimen Collection Point.
Logic Pack	A discrete software unit deployed with the server adaptor that encapsulates a set of related business processes. The logic pack contains data models, data formatting and parsing code, and business rules.
LSPN	Location Specific Practice Number.
MCAMT	Medicare Australia Middle Tier.
MDV	Medicare Data Validation.

Term/Acronym	Meaning
Mec	Medicare Easyclaim .
MPEC	Medical Patient Eligibility Check.
MT	Middle Tier.
Patient Claim	A claim for Medicare benefit made by or on behalf of a patient where the right to the benefit has not been assigned to the servicing practitioner.
PCe	Patient Claim Medicare Easyclaim – a message used in the transmitting of a patient claim in the Medicare Easyclaim system.
PeC	Patient Claming Medicare Easyclaim Cancel.
PCI	Patient Claim Interactive.
PLV	Practice Location Character.
Provider Number	A unique identifier, allocated by Medicare Australia to a medical practitioner for a specific medical practice.
PSM 2005	Protective Security Manual.
PVM	Patient Verification Medicare.
RBA	Reserve Bank of Australia.
SA	Server Adapter.
SCP	Specimen Collection Point (for pathology).
	Previously known as Licensed Collection Centre.
Server Adaptor	Software component provided to participants that wish to integrate their software in a pure-server role.
Specialist	Refers to Specialist/Dental/Allied Health
TCP/IP	Transmission Control Protocol / Internet Protocol – the basic communication protocol of the Internet. It provides a reliable connection oriented transport service. It can also be used as a communication protocol in a private network.
XML	Extensible Mark-up Language.
XML Schema	A standard from the World Wide Web Consortium for specifying XML data formats.

2 Medicare Easyclaim Logic Pack User Guide v2.0

This document is for use with the **Medicare Easyclaim Logic Pack v2.x** and is primarily aimed at developers implementing Medicare Easyclaim in a financial institution environment.

2.1 Overview

In August 2006, the Minister for Human Services, the Honourable Joe Hockey, announced that a facility will be made available to allow the submission of Medicare claims through the EFTPOS network, which will allow the Medicare benefit to be paid electronically to the claimant within 24 hours.

The Medicare Easyclaim solution will utilise the EFTPOS network to facilitate bulk bill and patient claiming with Medicare Australia.

Medicare Easyclaim is Medicare Australia's vehicle for delivering Business-to-Business (B2B) capability to financial institutions. Importantly, Medicare Easyclaim has evolved to accommodate new requirements, business requirements, policy, and to leverage contemporary technology, and will continue to do so over time.

Medicare Easyclaim (major version 2) contains the following business functions:

- Patient Claiming (PCe)
- Bulk Billing (BBe)

2.1.1 Patient Claiming (PCe)

The health care location will transmit claims through the EFTPOS system to a bank hub (BH), which will then forward the claim to Medicare Australia for processing. If the claim is successfully processed, the patient/claimant can provide their bank details (obtained through swiping their bank card through the EFTPOS device) and the bank will provide payment of the Medicare rebate on behalf of Medicare Australia to the claimant within 24 hours. Medicare Australia will pay the banking entity through the Reserve Bank of Australia (RBA).

The ability to perform patient claims in real-time was implemented with Online Claims in Release 1, early in 2001. This function allows a patient to claim from Medicare whilst at the providers practice. Medicare assesses the claim and provides a statement of claim and benefit back to the patient before the patient leaves the provider's practice.

2.1.2 Bulk Billing (BBe)

Bulk Billing (BBe) relates to services rendered by a medical practitioner to an eligible person (patient). Medicare pays benefits to persons (claimant/practitioner) who are eligible.

A practitioner who bulk bills undertakes to accept the relevant Medicare benefit as full payment for the service. The patient eligible for a benefit under the Medicare program assigns his or her right to the benefit to the practitioner. The practitioner then bills Medicare Australia instead of the patient.

2.2 What is a Logic Pack?

A logic pack is a set of Java classes that collectively describe a set of business processes that can be invoked using the Server Adaptor (SA).

These Java classes:

- for each business process, describe the structure of the various messages that are passed around within the Medicare Easyclaim architecture
- enforce the "simple" business rules associated with each message (for example, the syntax of fields, the completion of mandatory fields)
- bi-directionally translate each message between its Java "internal" representation and non-Java "external" representation (for example, XML)
- provide default values for target
- support dynamic discovery of the contents of the logic pack.

Each logic pack is packaged as a self-contained JAR file, with discovery (for instance, identifying which logic pack is contained within the JAR file) supported by the Java JAR service extension.

2.3 What is the Server Adaptor?

The Medicare Easyclaim Server Adaptor (eSA) is a software component provided to financial institutions participating in Medicare Easyclaim.

Key features of the server adaptor are:

- it is extended to include new business functionality using logic packs
- it is a Java web application that is designed to be re-configured and extended by the Medicare Easyclaim participant to meet site-specific requirements. *Out of the box*, its standard components are sufficient to implement basic connectivity.

2.4 Changes in R2.0

R2.0 introduces two new data elements:

a) *ServiceTypeCde* for BBe. This **mandatory** element allows the following values

- **O**: claim for general practitioner services
- **S**: claim for specialist services
- **D**: claim for diagnostic imaging services
- **P**: claim for pathology services

b) *RestrictiveOverrideCde*. This **optional** element allows the following values

- **SP**: ‘Separate Sites’ is a common term used to indicate the items were performed on separate sites of the body.
- **NR**: ‘Not Related’ occurs when a claim is being lodged and the provider needs to indicate that the item claimed is a consultation that is ‘Not Related’ to a Care Plan service. Where the ‘Not Related’ indicator is active the restrictive condition can be overridden to enable payment.
- **NC**: ‘Not for Comparison’ was an accepted value for *ItemOverrideCde* in R1

R2.0 modifies two existing data elements:

a) DateOfService

- Modified from no more than six months in the past to no more than two years in the past.

- b) SelfDeemedCde
  - Optional for BBe Pathology services

R2.0 introduces one new header constant:

a) hiconline.protocol.response.msgOrigin
- This header is set by the Medicare Middle Tier in response messages to show that the claim has reached the Medicare Middle Tier.

3 Patient Claims (PCe)

3.1 Patient Claiming Message Flow

Refer to Figure 1. Process Flow for Patient Claiming (PCe) below for an illustration of the Patient Claiming message flow.

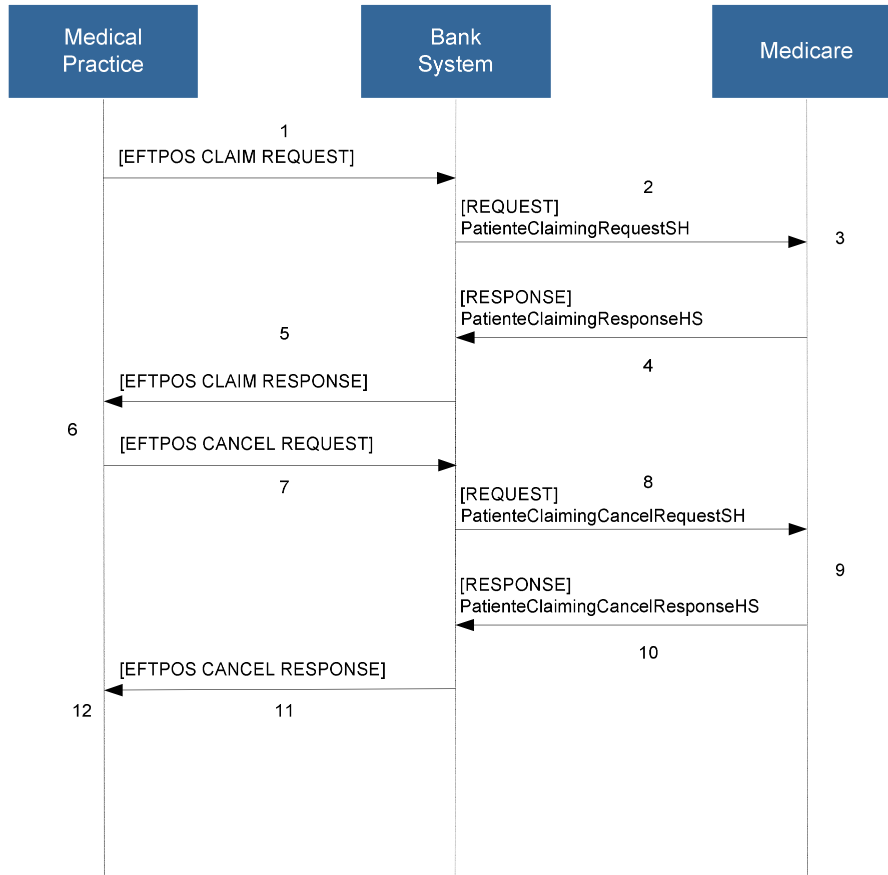

Figure 1. Process Flow for Patient Claiming (PCe)

3.1.1 Description of sequence PCe Flow

The main steps in the PCe process flow are as follows:

1. practice uses EFTPOS device to transmit the Patient Claim
2. Bank System (BS) uses the Server Adaptor to transmit the PatienteClaimingRequestSH to the Medicare Australia Hub
3. Medicare Australia performs a Medical Patient Eligibility Check (MPEC) and assesses the claim
4. a PatienteClaimingResponseHS is returned to the BS Server Adaptor
5. an EFTPOS Claim Response is returned to the EFTPOS device
6. the patient has the option to cancel the transaction (see steps 7-12) or the refund proceeds, and the EFTPOS device is used to print a receipt for the patient
7. practice uses EFTPOS to transmit the Cancel request
8. Bank System uses Server Adaptor to transmit the PatienteClaimingCancelRequestSH to the Medicare Australia Hub
9. Medicare Australia acknowledges the Patient Claim Cancel
10. a PatienteClaimingCancelRequestHS is returned to the Bank System
11. an EFTPOS Cancel Response is returned to the EFTPOS device
12. the EFTPOS device is used to print a receipt for the patient.

4 Bulk Billing (BBe)

4.1 Bulk Billing process Flow

Refer to Figure 2. Process Flow for Bulk Billing (BBe) below for an illustration of the Bulk Billing process flow.

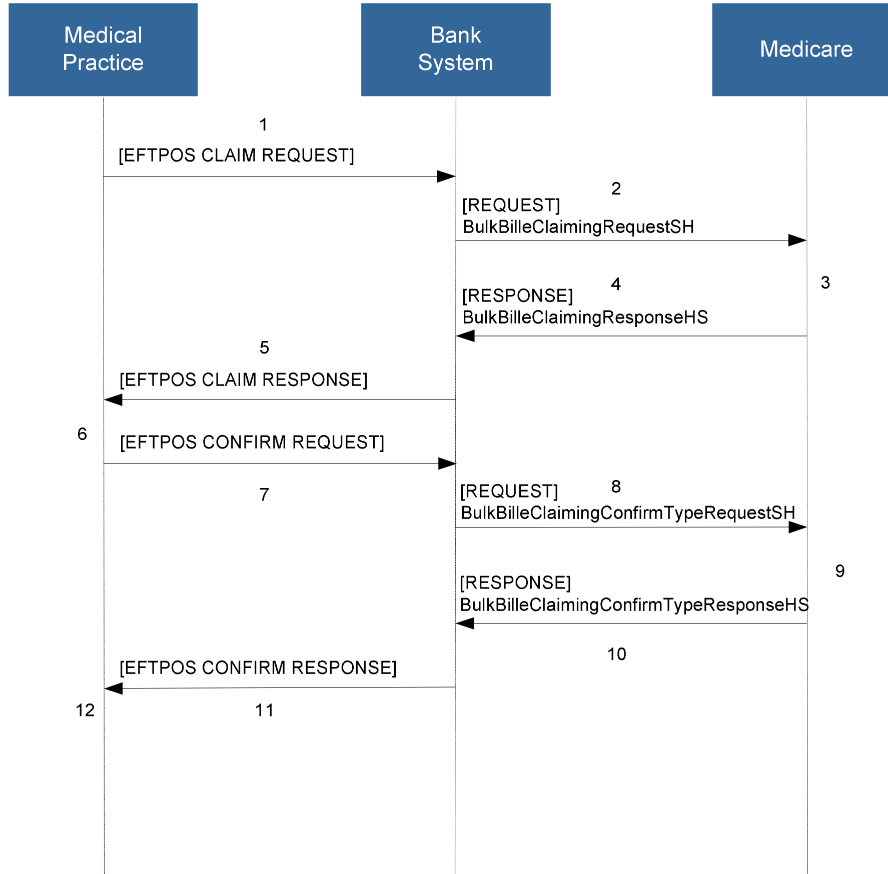

Figure 2. Process Flow for Bulk Billing (BBe)

4.1.1 Description of sequence BBe Flow

Synchronous BBe lodgement using the Server Adaptor:

1. practice uses EFTPOS device to transmit the Bulk Bill Claim
2. Bank System (BS) uses Server Adaptor to transmit the BulkBilleClaimingRequestSH to the Medicare Australia Hub
3. Medicare Australia performs a Medical Patient Eligibility Check (MPEC) and, if requested, a Concessional Entitlement Verification (CEV)
4. a BulkBilleClaimingResponseHS is returned to the BS Server Adaptor
5. an EFTPOS Claim Response is returned to the EFTPOS device
6. the provider accept/declines the Assignment of Benefit and the patient authorises/declines the Benefit Assignment
7. practice uses EFTPOS to transmit the confirm request
8. Bank System uses Server Adaptor to transmit the BulkBilleClaimingConfirmTypeRequestSH to the Medicare Australia Hub
9. Medicare Australia acknowledges the Bulk Bill Confirm
10. a BulkBilleClaimingConfirmTypeResponseHS is returned to the Bank System
11. an EFTPOS Confirm Response is returned to the EFTPOS device
12. the EFTPOS device is used to print a receipt for the patient.

5 Examples

The following examples detail how to:

- Generate a Request
- Validate a Request
- Generate XML Code
- Generate Java Object from XML

During this example please refer to:

- Appendix A Code Samples
- Appendix B Data Elements
- Appendix C Sample XML Messages
-

Refer to Figure 3. Logic Process Flow below for an illustration of the logic flow used in the following example.

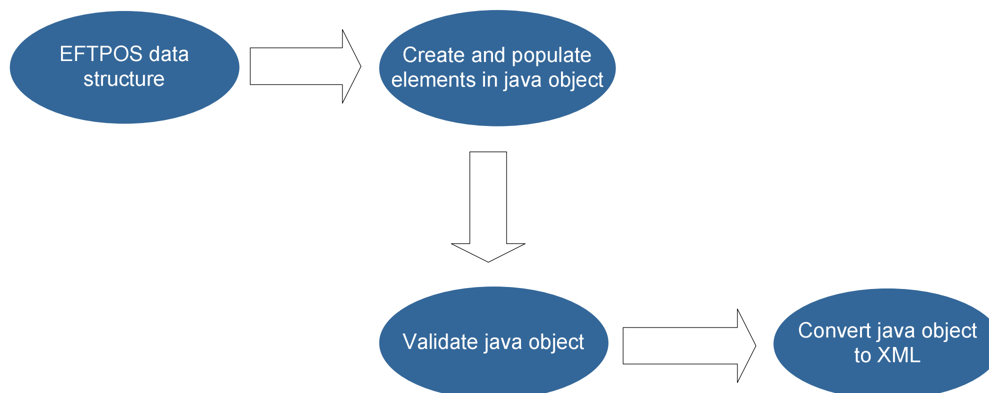

Figure 3. Logic Process Flow

5.1 Generate a Request

The following code generates a request:

```
public <<requestType>>Details createRequest()
{
        <<requestType>>Details request = new <<requestType>>Impl();
        // populate request attributes
        return request;
}
```

The `requestType` can be `PatienteClaimingRequest`, `PatienteClaimingCancelRequest`, `BulkBilleClaimingRequest`, `BulkBilleClaimingConfirmTypeRequest`.

A full list of code samples is provided in Appendix A Code Samples:

- A.1.1 Sample General Service Patient Claim Code
- A.1.2 Sample Specialist Service Patient Claim Code
- A.1.3 Sample Diagnostic Imaging Service Patient Claim Code
- A.1.4 Sample Cancel Request Patient Claim Code
- A.2.1 Sample code for General Practitioner Service
- A.2.2 Sample code for Specialist Service
- A.2.3 Sample code for Pathology Service
- A.2.4 Sample code for Diagnostic Imaging

5.2 Validate a Request

The following code validates a request:

```
public Vector validateRequest()
{
        Vector errors = new Vector();
        <<requestType>>Details request = new <<requestType>>Impl();
        // populate request attributes
        <<requestType>>Logic logic = new <<requestType>>Logic();
        logic.setLogicFor(request);
        logic.isValid(request,null,errors);
        return errors;
}
```

The `requestType` can be `PatienteClaimingRequest`, `PatienteClaimingCancelRequest`, `BulkBilleClaimingRequest`, `BulkBilleClaimingConfirmTypeRequest`.

A full list of data elements is provided in Appendix B Data Elements:

- B.1.1 Patient Claim (PCe) Request Data Elements
- B.1.2 Patient Claim (PCe) Cancel Request Data Elements
- B.1.3 Patient Claim (PCe) Response Data Elements
- B.1.4 Patient Claim (PCe) Cancel Response
- B.2.1 BulkBilleClaimingRequestSH Data Elements
- B.2.2 BulkBilleClaimingConfirmTypeRequestSH Data Elements
- B.2.3 BulkBilleClaimingResponseHS Data Elements

5.3 Generate XML Code

To convert from java to XML:

```
String xml = null;
try
{
        xml = BindManager.getInstance().toXml(
                request,
                <javaObjectClass>);
} catch (JAXBException e)
{
}
```

A full list of sample XML messages is provided in Appendix C Sample XML Messages.

- C.1.1 Sample PatienteClaimingRequestSH - General Practitioner Services XML Message
- C.1.2 Sample PatienteClaimingRequestSH - Specialist Services Request XML Message
- C.1.3

Sample PatienteClaimingRequestSH - Diagnostic Imaging Services Request XML Message

- C.1.4 Sample PatienteClaimingRequestSH - Not for Comparison - Restrictive override code NC XML Message
- C.1.5 Sample PatienteClaimingRequestSH - Separate Site - Restrictive override code SP XML Message
- C.1.6 Sample PatienteClaimingRequestSH - Not Related Care Plan and Consultation - Restrictive override code NR XML Message
- C.1.7 Sample PatienteClaimingResponseHS Response XML Message ExplanationCde (Length 3)
- C.1.8 Sample PatienteClaimingResponseHS Response XML Message ExplanationCde (Length 4)
- C.1.9 Sample PatienteClaimingCancelRequestSH Cancel Request XML Message
- C.1.10 Sample PatienteClaimingCancelResponseHS Cancel Response XML Message
- C.2.1 Sample Bulk Billing (BBe) General Practitioner Services Request XML Message
- C.2.2 Sample Bulk Billing (BBe) Specialist Services Request XML Message
- C.2.3 Sample Bulk Billing (BBe) Pathology Services Request XML Message
- C.2.4

Sample Bulk Billing (BBe) Diagnostic Imaging Services Request XML Message

- C.2.5 Sample BulkBilleClaimingConfirmTypeRequestSH Confirm Type Request XML Message
- C.2.6 Sample BulkBilleClaimingResponseHS Response XML Message
- C.2.7 Sample BulkBilleClaimingConfirmTypeResponseHS Confirm Type Response XML Message

5.4 Generate Java Object from XML

To convert from XML to java:

```
try
{
      Object rval =
            BindManager.getInstance().fromXml(
                  xmlMessage,
                  <javaObjectClass>);
      return rval;
} catch (JAXBException e)
{
}
```

6 How to use the response

To use the response from the Medicare Easyclaim Server Adaptor:

- Retrieve all of the headers returned by the Medicare Easyclaim Server Adaptor.
- Check the value returned in the header:
  - `hiconline.protocol.remote.statuscode`
- In most cases, if the value is **'0'**, then the XML is returned in the body. This XML needs to be used to send the appropriate response to the EFTPOS device.
- Normally a non-zero value indicates an error condition. The exception to this is when **'1716'** is returned in a Bulk Billed Confirm Response, where it indicates the claim has been received by Medicare Australia and is being processed.

No.	Request Type	Status Code for successful transmission	Response XML present
1	PCe	0	Yes
2	PeC	0	No
3	BBe	0	Yes
4	BeC (Accept)	1716	No
5	BeC (Decline)	0	No

Table 1. Request Type, Status Code, and XML Response

7 Return Codes

Refer to Table 2. Return Codes below for a list of return codes and their explanations.

Code	Explanation
2015	No voucher exists within the session for the supplied VoucherSeqNum.
2016	No service exists in the claim for the supplied service ID.
2017	The Payee Provider specified is the same as the Servicing Provider.
2025	The maximum number of child business objects for the parent business object type has been reached.
2030	The data element being set is inconsistent with other data elements already set OR a data element has been set and a related conditionally required data element has not been set.
2032	The maximum number of services allowable for the voucher has been reached.
2038	The referral/request type is inconsistent with the service type set for this claim.
2053	Patient contribution amount must be less than total charge.
2055	Patient contribution amount should not be set when the account is fully paid.
3998	First three characters of eClaiming transaction id should be Financial Institution ID.
9006	The Provider is not authorised to participate in Online Claiming. Contact the Medicare Australia eBusiness Service Centre for further assistance.
9142	The value in the Restrictive Override Code is invalid, please check and resubmit your claim
9201	Invalid format for data item.
9203	Date of service must be no more than six (6) months in the past.
9204	Date in future. The date supplied must not be in the future.
9207	An item cannot be self deemed or substituted when a referral or request override has been set.
9210	Date of service must be no more than two years in the past
9301	Patient's Medicare card number must be supplied.
9302	Patient's reference number must be supplied.
9305	Servicing Practitioner/Payee Practitioner Provider Number must be supplied.
9306	Date of service must be supplied.
9307	An item number must be supplied for each service.
9308	Referring Practitioner's Provider Number must be supplied.
9310	Requesting Practitioner's Provider Number must be supplied.
9311	Request issue date must be supplied, and must be prior to, or the same as, the date of the medical service and cannot be before the date of birth.
9312	Claimant first name, family name, date of birth, claimant Medicare card number and reference number must be supplied. If any one data element is supplied, then all five (5) must be supplied.
9316	The Referring/Requesting Provider cannot be the Servicing or Principal Provider.
9317	Account payment status required. Must be paid or unpaid.
9322	Referral period details must be supplied.
9325	Service type not supplied.
9326	At least one voucher must be included in the claim.
9332	Voucher must contain at least one (1) service.
9338	A required charge amount has not been supplied or is inconsistent with other data supplied.
9364	Patient information provided is insufficient.
9427	Service start date must be on or after the patient’s date of birth and on or before the date of service and service end date.
9601	The claim needs to be referred to a Medicare assessor
9602	This claim cannot be paid through Online Claiming. Cancel the claim and print an account/receipt for the claimant to submit direct to Medicare.
9605	Check Patient details. Another Medicare Card may have been issued to the patient or the details you hold do not match those held by Medicare. Please update your records and resubmit the claim.
9606	Check Claimant details. Another Medicare Card may have been issued to the claimant or the details you hold do not match those held by Medicare. Please update your records and

	resubmit the claim.
9611	Check item. The item claimed is either unknown or invalid at the date of service.
9618	Check charge
9625	A claimant address is required for this claim
9630	Check request/referral details
9631	Check if service self deemed
9632	Duplicate of service already paid
9635	Check Servicing Provider
9638	Claimant details required. Patient or quoted claimant is a minor.
9641	A restrictive condition exists
9643	Check claimant name
9646	The claim could not be located by Medicare Australia.
9655	An LSPN is required
9661	This Provider cannot substitute services
9678	The service is not payable as an appropriate associated service is not present
9682	Medicare cannot assess this request due to a system limitation. Please contact the Medicare Australia eBusiness Service Centre for assistance.
9698	Service is possible aftercare, check the account and resubmit with a valid indicator if not normal aftercare.
9699	Item not covered for this patient at this date of service
9700	An incorrect item number appears to have been used or amount claimed does not match item number
9701	The maximum number of services for this item have been paid, if this service is not a duplicate please resend with correct item numbers as per MBS
9702	A base item has not been entered or should be entered first. Please re-submit claim with correct sequence.
9703	Item number used can not be claimed for this Provider. Check details of service and re-submit with appropriate item.
9704	This service appears to have been previously claimed. Please contact Medicare if you wish to discuss.
9705	In some circumstances where two or more services are performed together, they are claimable under one item number. Please check the MBS and re-submit.
9706	This item requires a specific notation of the relevant condition. Please check the MBS and resubmit through an alternative Medicare claim channel.
9707	This claim needs to be referred to a Medicare Customer Services Officer for further assessment. Please issue claimant with an account/receipt to claim with Medicare through alternative channel.
9708	Equipment number entered does not appear to be registered with Medicare Australia, correct details and re-submit or contact Medicare.
9709	An age restriction applies to this item. Please check the MBS to verify item specifics.
9710	This item number has specific restrictions. Please refer to the MBS and ensure you are entering the correct patient details.
9711	This claim requires further assessment by a Medicare Customer Services Officer. Please issue claimant with an account/receipt to claim with Medicare through alternative channel
9712	The item number claimed and an override code used cannot be used together. Please resubmit the claim or contact Medicare Australia for assistance.
9765	Site not accredited for this service.

Table 2. Return Codes

Appendix A. Code Samples

A.1. Sample Patient Claim (PCe) Code

There are three types of Patient Claims:

- General Service
- Specialist
- Diagnostic

Each type is shown in the following sections.

A.1.1. Sample General Service Patient Claim Code

The following is a sample code for a General Service Patient Claim:

PCe General Service Claim

```
public PatienteClaimingRequestDetails createRequestDetails()
   {
     //PCe General Service Request

     PatienteClaimingRequestDetails request =
        new PatienteClaimingRequestDetailsImpl();

     PCeClaimDetails claim = new PCeClaimDetailsImpl();

     MembershipDetails claimant = new MembershipDetailsImpl();
     claimant.setMemberNum("2147712345");
     claimant.setMemberRefNum("1");

     MembershipDetails patient = new MembershipDetailsImpl();
     patient.setMemberNum("2147712345");
     patient.setMemberRefNum("3");

     ProviderDetails servicing = new ProviderDetailsImpl();
     servicing.setProviderNum("5555555W");

     claim.setAccountPaidInd("Y");
           claim.setClaimant(claimant);
           claim.setPatient(patient);
           claim.setServicingProvider(servicing);

     PCeVoucherDetails voucher = new PCeVoucherDetailsImpl();
           voucher.setVoucherId("01");
           voucher.setServiceTypeCde("O");

     PCeServiceDetails service = new PCeServiceDetailsImpl();
     service.setServiceId("0001");

     service.setChargeAmount("5000");

     Calendar cal = Calendar.getInstance();
     cal.add(Calendar.DATE, -1);
           service.setDateOfService(cal);

     service.setMbsItemNum("000023");

           claim.addVoucher(voucher);
           voucher.addService(service);

           request.setClaim(claim);

     return request;
    }
```

A.1.2. Sample Specialist Service Patient Claim Code

The following is a sample code for a Specialist Patient Claim:

```
PCe Specialist Claim
public PatienteClaimingRequestDetails createRequestDetails()
    {
     PatienteClaimingRequestDetails request =
        new PatienteClaimingRequestDetailsImpl();

     //PCe Specialist Request

     PCeClaimDetails claim = new PCeClaimDetailsImpl();

     MembershipDetails claimant = new MembershipDetailsImpl();
     claimant.setMemberNum("2147712345");
     claimant.setMemberRefNum("1");

     MembershipDetails patient = new MembershipDetailsImpl();
     patient.setMemberNum("2147712345");
     patient.setMemberRefNum("3");

     ProviderDetails servicing = new ProviderDetailsImpl();
     servicing.setProviderNum("5555555W");

     claim.setAccountPaidInd("Y");
          claim.setClaimant(claimant);
          claim.setPatient(patient);
          claim.setServicingProvider(servicing);

     PCeVoucherDetails voucher = new PCeVoucherDetailsImpl();
     voucher.setVoucherId("01");

     ReferralDetails referral = new ReferralDetailsImpl();

     Calendar calRef = Calendar.getInstance();
     calRef.add(Calendar.DATE, -1);
          referral.setDateOfIssue(calRef);
     referral.setPeriodTypeCde("S");

     ProviderDetails referring = new ProviderDetailsImpl();
     referring.setProviderNum("5555555W");
          referral.setProvider(referring);

          voucher.setReferral(referral);
     voucher.setServiceTypeCde("S");

     PCeServiceDetails service = new PCeServiceDetailsImpl();
     service.setServiceId("0001");
     service.setChargeAmount("10000");

     Calendar cal = Calendar.getInstance();
     cal.add(Calendar.DATE, -1);
          service.setDateOfService(cal);

     service.setMbsItemNum("000105");

          claim.addVoucher(voucher);
          voucher.addService(service);

          request.setClaim(claim);

     return request;
    }
```

A.1.3. Sample Diagnostic Imaging Service Patient Claim Code

The following is a sample for a Diagnostic Imaging Patient Claim:

```
PCe Diagnostic Imaging Claim

public PatienteClaimingRequestDetails createRequestDetails()
    {
```

```
    PatienteClaimingRequestDetails request =
      new PatienteClaimingRequestDetailsImpl();

    //PCe Diagnostic Request

    PCeClaimDetails claim = new PCeClaimDetailsImpl();

    MembershipDetails claimant = new MembershipDetailsImpl();
    claimant.setMemberNum("2147712345");
    claimant.setMemberRefNum("1");

    MembershipDetails patient = new MembershipDetailsImpl();
    patient.setMemberNum("2147712345");
    patient.setMemberRefNum("3");

    ProviderDetails servicing = new ProviderDetailsImpl();
    servicing.setProviderNum("5555555W");

    claim.setAccountPaidInd("Y");
         claim.setClaimant(claimant);
         claim.setPatient(patient);
         claim.setServicingProvider(servicing);

    PCeVoucherDetails voucher = new PCeVoucherDetailsImpl();
    voucher.setVoucherId("01");

    RequestDetails req = new RequestDetailsImpl();

    Calendar calRef = Calendar.getInstance();
    calRef.add(Calendar.DATE, -1);
         req.setDateOfIssue(calRef);
    req.setTypeCde("D");

    ProviderDetails requesting = new ProviderDetailsImpl();
    requesting.setProviderNum("5555555W");
         req.setProvider(requesting);

         voucher.setRequest(req);
    voucher.setServiceTypeCde("D");

    PCeServiceDetails service = new PCeServiceDetailsImpl();
    service.setServiceId("0001");
    service.setChargeAmount("40000");

    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.DATE, -1);
         service.setDateOfService(cal);

    service.setLspnNum("111234");
    service.setMbsItemNum("057041");

         claim.addVoucher(voucher);
         voucher.addService(service);

         request.setClaim(claim);

    return request;
   }
```

A.1.4. Sample Cancel Request Patient Claim Code

The following is a sample for a Patient Claim (PCe) Cancel Request:

```
PCe Cancel Request
public PatientClaimeClaimingDeclineTypeRequestDetails createCancelReqDetails()
           {
```

```
        PatientClaimeClaimingDeclineTypeRequestDetails request =
           new PatientClaimeClaimingDeclineTypeRequestDetailsImpl();

        request.setAcceptInd("N");

        return request;
           }
```

A.2. Sample Bulk Billing (BBe) Request Code

A.2.1. Sample code for General Practitioner Service

The following is sample code for a General Practitioner Service:

```
General Practitioner Service

  public BulkBilleClaimingRequestDetails createRequestDetails()
  {
    BulkBilleClaimingRequestDetails request =
              new BulkBilleClaimingRequestDetailsImpl();

    BBeClaimDetails claim = new BBeClaimDetailsImpl();
    BBeVoucherDetails voucher = new BBeVoucherDetailsImpl();
    voucher.setVoucherId("01");
    voucher.setServiceTypeCde("O");
    BBeServiceDetails service = new BBeServiceDetailsImpl();
    service.setServiceId("0001");

    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.DATE, -1);

    MembershipDetails patient = new MembershipDetailsImpl();
    patient.setMemberNum("2147712345");
    patient.setMemberRefNum("1");

    ProviderDetails servicing = new ProviderDetailsImpl();
    servicing.setProviderNum("5555555W");

    claim.setCevRequestInd("Y");
    claim.setPatient(patient);
    claim.setServicingProvider(servicing);

    service.setDateOfService(cal);
    service.setMbsItemNum("000053");

    request.setClaim(claim);
    claim.addVoucher(voucher);
    voucher.addService(service);

    return request;
  }
```

A.2.2. Sample code for Specialist Service

The following is sample code for a Specialist Service:

Specialist Service

```
public BulkBilleClaimingRequestDetails createRequestDetails()
{
  BulkBilleClaimingRequestDetails request =
             new BulkBilleClaimingRequestDetailsImpl();

  BBeClaimDetails claim = new BBeClaimDetailsImpl();
  BBeVoucherDetails voucher = new BBeVoucherDetailsImpl();
  voucher.setVoucherId("01");
  voucher.setServiceTypeCde("S");
  BBeServiceDetails service = new BBeServiceDetailsImpl();
  service.setServiceId("0001");

  Calendar cal = Calendar.getInstance();
  cal.add(Calendar.DATE, -1);

  MembershipDetails patient = new MembershipDetailsImpl();
  patient.setMemberNum("2147712345");
  patient.setMemberRefNum("1");

  ProviderDetails payee = new ProviderDetailsImpl();
  payee.setProviderNum("9999999F");

  ProviderDetails servicing = new ProviderDetailsImpl();
  servicing.setProviderNum("5555555W");

  claim.setPatient(patient);
  claim.setPayeeProvider(payee);
  claim.setServicingProvider(servicing);

  ProviderDetails referralProvider = new ProviderDetailsImpl();
  referralProvider.setProviderNum("0000071W");

  ReferralDetails referral = new ReferralDetailsImpl();
  referral.setProvider(referralProvider);
  referral.setPeriodTypeCde("S");
  referral.setDateOfIssue(cal);

  voucher.setReferral(referral);

  service.setDateOfService(cal);
  service.setMbsItemNum("000105");

  request.setClaim(claim);
  claim.addVoucher(voucher);
  voucher.addService(service);

  return request;
}
```

A.2.3. Sample code for Pathology Service

The following is sample code for a Pathology Service:

Pathology Service

```
  public BulkBilleClaimingRequestDetails createRequestDetails()
  {
    BulkBilleClaimingRequestDetails request =
               new BulkBilleClaimingRequestDetailsImpl();

    BBeClaimDetails claim = new BBeClaimDetailsImpl();
    BBeVoucherDetails voucher = new BBeVoucherDetailsImpl();
    voucher.setVoucherId("01");
    voucher.setServiceTypeCde("P");
    BBeServiceDetails service = new BBeServiceDetailsImpl();
    service.setServiceId("0001");

    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.DATE, -1);

    MembershipDetails patient = new MembershipDetailsImpl();
    patient.setMemberNum("2147712345");
    patient.setMemberRefNum("1");

    ProviderDetails payee = new ProviderDetailsImpl();
    payee.setProviderNum("9999999F");

    ProviderDetails servicingProvider = new
ProviderDetailsImpl();
    servicingProvider.setProviderNum("5555555W");

    claim.setCevRequestInd("Y");
    claim.setPatient(patient);
    claim.setPayeeProvider(payee);
    claim.setServicingProvider(servicingProvider);

    ProviderDetails requestProvider = new ProviderDetailsImpl();
    requestProvider.setProviderNum("0000071W");

    RequestDetails req = new RequestDetailsImpl();
    req.setProvider(requestProvider);
    req.setTypeCde("P");
    req.setDateOfIssue(cal);

    voucher.setRequest(req);

    service.setDateOfService(cal);
    service.setMbsItemNum("065120");
    service.setScpIdNum("0278");

    request.setClaim(claim);
    claim.addVoucher(voucher);
    voucher.addService(service);

    return request;
  }
```

A.2.4. Sample code for Diagnostic Imaging

The following is sample code for a Diagnostic Imaging:

```
Diagnostic Imaging

  public BulkBilleClaimingRequestDetails createRequestDetails()
  {
    BulkBilleClaimingRequestDetails request =
               new BulkBilleClaimingRequestDetailsImpl();

    BBeClaimDetails claim = new BBeClaimDetailsImpl();
    BBeVoucherDetails voucher = new BBeVoucherDetailsImpl();
    voucher.setVoucherId("01");
    voucher.setServiceTypeCde("D");
    BBeServiceDetails service = new BBeServiceDetailsImpl();
    service.setServiceId("0001");

    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.DATE, -1);

    MembershipDetails patient = new MembershipDetailsImpl();
    patient.setMemberNum("2147712345");
    patient.setMemberRefNum("1");

    ProviderDetails payee = new ProviderDetailsImpl();
    payee.setProviderNum("9999999F");

    ProviderDetails servicingProvider = new
ProviderDetailsImpl();
    servicingProvider.setProviderNum("5555555W");

    claim.setCevRequestInd("Y");
    claim.setPatient(patient);
    claim.setPayeeProvider(payee);
    claim.setServicingProvider(servicingProvider);

    ProviderDetails requestProvider = new ProviderDetailsImpl();
    requestProvider.setProviderNum("0000071W");

    RequestDetails req = new RequestDetailsImpl();
    req.setProvider(requestProvider);
    req.setTypeCde("D");
    req.setDateOfIssue(cal);

    voucher.setRequest(req);

    service.setDateOfService(cal);
    service.setMbsItemNum("058100");
    service.setLspnNum("7874");

    request.setClaim(claim);
    claim.addVoucher(voucher);
    voucher.addService(service);

    return request;
  }
```

A.2.5. Sample code for Confirm Type Request

The following is sample code for a Confirm Type Request:

Confirm Type Request

```
public BulkBilleClaimingConfirmTypeRequestDetails createRequestDetails()
    {
          BulkBilleClaimingConfirmTypeRequestDetails request =
                                     new
BulkBilleClaimingConfirmTypeRequestDetailsImpl();

          ConfirmTypeRequestDetails confirm = new
ConfirmTypeRequestDetailsImpl();
          confirm.setBenefitAssignmentAuthorisedInd("Y");
          confirm.setConfirmTypeInd("Y");

          request.setConfirm(confirm);

          return request;
    }
```

Appendix B. Data Elements

This appendix contains tables listing the various Medicare Easyclaim data elements.

B.1. Patient Claim (PCe) Data Elements

B.1.1. Patient Claim (PCe) Request Data Elements

In Table 3. Patient Claiming (PCe) Request Data Elements below the **Bold** text lists the **`Element Name`** and the *Italicized* text represents the *`Schema Path`*. In the M/C/O column **M = M**andatory, **C = C**onditional, and **O = O**ptional, while in the Type column **A**=**A**lpha, **N**=**N**umeric, **D**=**D**ate, **AN**=**A**lpha**N**umeric

Element Name	Size	M/C/O			Type	Rule/Validation
		General	Specialist	D.Imaging		
Transmission Data Elements as per MeC Technical Specification						
Claim Data Elements (PCeClaim)						
AccountReferenceId *accountReferenceNum*	9	O	O	O	AN	▪ Can have up to 9 AN characters. ▪ Invalid format/value [9201] ▪ **Note**: Only relevant for unpaid accounts
AccountPaidInd *accountPaidInd > IndicatorEnum*	1	M	M	M	A	Valid values ▪ **Y** = Account fully paid ▪ **N** = Account not fully paid Invalid format/value [9201] If not set [9317]
ClaimantMedicareCardNum *claimant > memberNum*	10	C	C	C	N	▪ Condition: Must be set if other claimant details are set [9312] ▪ Must conform to the Medicare card check digit routine [9201]
ClaimantReferenceNum *claimant > memberRefNum*	1	C	C	C	N	▪ Valid values: 1 – 9 Condition: Must be set if other claimant details are set [9312]
PatientMedicareCardNum *patient > memberNum*	10	M	M	M	N	▪ Must conform to the Medicare card check digit routine [9201] ▪ Must be set if other patient details are set [9364]
PatientReferenceNum *patient > memberRefNum*	1	M	M	M	N	▪ Valid values: 1 – 9 [9201] ▪ Must be set if other patient details are set [9364]
PayeeProviderNum *payeeProvider > providerNum*	8	O	O	O	AN	▪ Must conform to the Medicare provider check digit routine [9201] ▪ Must not be the same stem (first 6 numbers) as the ServicingProviderNum. Matching stems [2017]
ServicingProviderNum *servicingProvider > providerNum*	8	M	M	M	AN	▪ Must conform to the Medicare provider check digit routine [9201] ▪ Must not be the same stem (first 6 numbers) as the PayeeProviderNum. Matching stems [2017] ▪ If not set returns [9305]
Second level Data Elements (PCeVoucher)						
VoucherId *voucherId*	2	M	M	M	N	▪ Invalid format/value or not set [9201] ▪ If not set [9326] ▪ If greater than max vouchers (1) [2025]

Element Name	Size	M/C/O General	M/C/O Specialist	M/C/O D.Imaging	Type	Rule/Validation
ReferralIssueDate *referral > dateOfIssue*	8	-	C	-	D	Condition: cannot be set if ReferralOverrideTypeCde is set [2030] ▪ Must be a valid date Invalid format/value [9201] ▪ Cannot be after the DateOfService [2030] ▪ If not set [9309] ▪ Cannot be a in the future [9204]

ReferralPeriodTypeCde
referral > periodTypeCde
	1	-	C	-	A	Condition: cannot be set if ReferralOverrideTypeCde is set [2030]
						▪ If not set [2038]
						▪ Valid values:
						  ▪ **S** = Standard
						  ▪ **N** = Non standard
						  ▪ **I** = Indefinite
						Invalid format/value or not set [9201]

ReferralOverrideTypeCde
referralOverrideTypeCode
	1	-	O	-	A	Valid values:
						▪ **L** = Lost
						▪ **E** = Emergency
						▪ **N** = Not required
						Invalid format/value or not set [9201]
						▪ Can only be used when ServiceTypeCde, if applicable to the claim type, is set **S** [2030]
						▪ Mutually exclusive to:
						  ▪ ReferringProviderNum
						  ▪ ReferralTypeCde
						  ▪ ReferralIssueDate
						  [2030]

ReferringProviderNum *referral > providerNum*	8	-	C	-	AN	Condition: cannot be set if ReferralOverrideTypeCde is set [2030] ▪ Cannot be the same as ServicingProviderNum [9316] ▪ If not set [9308] ▪ Must conform to the Medicare provider check digit routine [9201]
RequestIssueDate *Request > dateOfIssue*	8	-	-	C	D	Condition: cannot be set if RequestOverrideTypeCde is set [2030] ▪ Must be a valid date ▪ Invalid format/value [9201] ▪ Cannot be a in the future [9204] ▪ Cannot be after the DateOfService [9311] ▪ If not set [9311]

RequestOverrideTypeCde
requestOverrideTypeCde
	1	-	-	O	A	Valid values:
						▪ **L** = Lost
						▪ **E** = Emergency
						▪ Invalid format/value [9201]
						▪ Mutually exclusive to
						  ▪ RequestingProviderNum
						  ▪ RequestTypeCde
						  ▪ RequestIssueDate
						  ▪ [2030]

Element Name	Size	M/C/O			Type	Rule/Validation
		General	Specialist	D.Imaging		
RequestTypeCde *request > typeCde > RequestTypeCdeEnum*	1	-	-	C	A	▪ Condition: cannot be set if RequestOverrideTypeCde is set [2030] Valid values: ▪ **D** = Diagnostic Imaging request [9201] ▪ If not set [2030]
RequestingProviderNum *Request > providerNum*	8	-	-	C	AN	Condition: cannot be set if RequestOverrideTypeCde is set [2030] ▪ Cannot be the same as ServicingProviderNum [9316] ▪ Must conform to the Medicare provider check digit routine [9201] ▪ If not set [9310]
ServiceTypeCde *ServiceTypeCde > ServiceTypeCdeEnum*	1	M	M	M	A	Valid values ▪ **O** = General Practitioner ▪ **S** = Specialist ▪ **D** = Diagnostic Imaging Invalid format/value [9201] When **O,** no request elements and no referral elements can be set [2030] When **S,** referral elements should be populated and only provided (no request elements) [2030] **OR** only referralOverrideTypeCde [2030] When **D,** request elements should be populated and only provided (no referral elements) [2030] **OR** only requestOverrideTypeCde [2030] **OR** only SelfDeemed ='SD' [9207]. When not set [9325]
Service Data Elements (PCeService)						
ServiceId *serviceId*	4	M	M	M	N	▪ Valid values 0001-0014 [9201] ▪ Invalid format/value [9201] ▪ If not set [9322] Greater than max services [2032]
ChargeAmount *chargeAmount*	7	M	M	M	N	In cents i.e. 999999 is $9999.99 ▪ Minimum value of 100 ($1.00) ▪ Notional charging is not allowed ▪ Invalid format/value [9201] ▪ If not set [9338] ▪ **Note**: Should be rounded to the nearest five cents
DateOfService *dateOfService*	8	M	M	M	D	▪ Cannot be a date in the future [9204] ▪ Cannot be more than 2 years in the past [9210] ▪ Invalid format/value [9201] ▪ If not set [9306]
ItemNum *mbsItemNum*	6	M	M	M	AN	▪ Invalid format/value [9201] ▪ If not set [9307] ▪ **Note**: If ItemNum is 6 characters and does not start with "0" it will be rejected in the Medicare Hub [9611]

ItemOverrideCde
itemOverrideCde
	2	O	O	O	AN	▪ Invalid format/value [9201]
						▪ Valid values:
						  ▪ **AP** = Not duplicate service
						  ▪ **AO** = Not normal aftercare
						▪ **Note:** NC removed as a valid value in R2

Element Name	Size	M/C/O General	M/C/O Specialist	M/C/O D.Imaging	Type	Rule/Validation
RestrictiveOverrideCde *restrictiveOverrideCde*	2	O	O	O	AN	▪ Invalid value for Restrictive Override [9142] ▪ Valid values: ▪ **SP** = Separate sites ▪ **NR** = Not related ▪ **NC** = Not for comparison
PatientContribAmt *patientContribAmt*	7	C	C	C	N	In cents i.e. 999999 is $9999.99 ▪ Minimum value of 100 ($1.00) ▪ Must be less than ChargeAmount [2053] ▪ Can only be set if AccountPaidInd = N [2055] ▪ Invalid format/value [9201] ▪ **Note**: Should be rounded to the nearest five cents
EquipmentId *equipmentIdNum*	5	-	-	O	AN	▪ If set, LSPNum must be present [2030] ▪ Cannot be set to, or equal, zero ▪ Invalid format/value [9201] Mutually exclusive to: ▪ ReferralPeriodTypeCde ▪ ReferringProviderNum ▪ ReferralIssueDate [2030] • **Note**: Only required for Radiation Oncology
SelfDeemedCde *selfDeemedCde > selfDeemedCdeEnum*	2	-	-	C	A	Condition: If set to SS, Request details must also be supplied [2030] ▪ Cannot be set to SD if the containing voucher details a Request [2030] ▪ Valid values: ▪ **SD** = Self Deemed ▪ **SS** = Substituted Service ▪ **N** = Neither Self Deemed or Substituted Invalid format/value [9201] Mutually exclusive to: ▪ ReferralOverrideTypeCde ▪ RequestOverrideTypeCde ▪ [or 9207 is returned] ▪ ReferralIssueDate ▪ ReferralPeriodTypeCde ▪ ReferringProviderNum ▪ [or 2030 is returned]
LSPNum *lspNum*	6	O	O	C	N	▪ Cannot be set to 000000 [9201] ▪ Invalid format/value [9201] Must be set if EquipmentId is set [2030]

Table 3. Patient Claiming (PCe) Request Data Elements

B.1.2. Patient Claim (PCe) Cancel Request Data Elements

In Table 4. Patient Claim (PCe) Cancel below the **Bold** text lists the **`Element Name`** and the *Italicized* text represents the *`Schema Path`*. In the M/C/O column **M = M**andatory, **C = C**onditional, and **O = O**ptional, while in the Type column **A**=**A**lpha, **N**=**N**umeric, **D**=**D**ate, **AN**=**A**lpha**N**umeric.

Element Name/Path	Size	M/C/O	Type	Notes, Format, Values, Constraints
PCeCancel				
AcceptIndicator *acceptInd*	1	M	AN	Value must be set to 'N'. Invalid format/value [9201] If not set [2030]

Table 4. Patient Claim (PCe) Cancel

B.1.3. Patient Claim (PCe) Response Data Elements

In Table 5. Patient Claim (PCe) Response below the **Bold** text lists the **`Element Name`** and the *Italicized* text represents the *`Schema Path`*. In the M/C/O column **M = M**andatory, **C = C**onditional, and **O = O**ptional, while in the Type column **A**=**A**lpha, **N**=**N**umeric, **D**=**D**ate, **AN**=**A**lpha**N**umeric.

Element Name	Size	M/C/O	Type
Transmission Data Elements as per MeC Technical Specification			
PCeClaimResponse			
AccountPaidInd *accountPaidInd > IndicatorEnum*	1	M	A
AcceptanceTypeCde *medicareAcceptanceType*	4	M	AN
AssessmentError *assessmentError*	4	C[1]	N
AssessmentStatus *assessmentStatus*	4	M	A
ClaimantFirstName *claimant > identity > firstName*	40	C[2]	ANS
ClaimantLastName *claimant > identity > lastName*	40	C[2]	ANS
ClaimantMedicareCardNum *claimant > membership > currentMedicareCardNum*	10	C[2]	N
ClaimantReferenceNum *claimant > membership> currentSubnumerate*	1	C[2]	N
PatientFirstName *patient > identity > firstName*	40	M	ANS
PatientLastName *patient > identity > lastName*	40	M	ANS
PatientMedicareCardNum *patient > currentMedicareCardNum*	10	M	N
PatientReferenceNum *patient > currentSubnumerate*	1	M	N

Element Name	Size	M/C/O	Type
ProviderNum *providerResponse> provider > providerNum*	8	M	AN
ProviderName *providerResponse > name*	27	M	ANS
PCeVoucherResponse			
VoucherId *voucherId*	2	M	N
PCeServiceResponse			
ServiceId *serviceId*	4	M	N
ChargeAmount *chargeAmount*	7	M	N
DateOfService *dateOfService*	8	M	D
ItemNum *mbsItemNum*	6	M	N
PatientContribAmt *patientContribAmt*	7	M	N
ScheduleFee *scheduleFee*	7	M	N
ServiceBenefitAmount *BenefitAmount*	7	M	N
AssessmentStatus *assessmentStatus*	4	M	A
ExplanationCde *assessmentExplanationCde > Explanation > code*	4[3]	O	AN
ExplanationText *assessmentExplanationText > Explanation > text*	50	O	ANS

Table 5. Patient Claim (PCe) Response

Table Notes:
[1] - Condition - For a failed claim only.
[2] - Condition - Only for unpaid, partially paid, or pended patient claims
[3] – For successfully paid claims, the field will contain a 3 character code. For rejected claims, this field will contain a 4 digit error code. Refer to appendix C 1.1.7 Sample PatienteClaimingResponseHS Response XML Message ExplanationCde (Length 3) and appendix C 1.1.8 Sample PatienteClaimingResponseHS Response XML Message ExplanationCde (Length 4).

B.1.4. Patient Claim (PCe) Cancel Response

There are no Data Elements in a Patient Claim (PCe) Cancel Response message. There is only the Header, the Content of the message is empty.

B.2. Bulk Bill (BBe) Data Elements

B.2.1. BulkBilleClaimingRequestSH Data Elements

In Table 6. BulkBilleClaimingRequestSH below the **Bold** text lists the **`Element Name`** and the *Italicized* text represents the *`Schema Path`*. In the M/C/O column **M = M**andatory, **C = C**onditional, and **O = O**ptional, while in the Type column **A**=**A**lpha, **N**=**N**umeric, **D**=**D**ate, **AN**=**A**lpha**N**umeric.

Element Name	Size	M/C/O				Type	Rule/Validation
		General	Specialist	Pathology	D.Imaging		
Transmission Data Elements as per MeC Technical Specification							
Claim Data Elements (BBeClaim)							
PatientMedicareCardNum *patient > memberNum*	10	M	M	M	M	N	▪ Must conform to the Medicare card check digit routine [9201] ▪ If not set [9364]
PatientReferenceNum *patient > memberRefNum*	1	M	M	M	M	N	▪ Valid values: 1 – 9 [9201] ▪ If not set [9364]
ServicingProviderNum *servcingProvider > providerNum*	8	M	M	M	M	AN	▪ Must conform to the Medicare provider check digit routine [9201] ▪ If not set [9305]

CEVRequestInd
cevRequest
	1	M	-	M	M	ANS	▪ Valid values:
							  ▪ **Y** = Yes
							  ▪ **N** = No [9201]
							▪ If not set [2030]

PayeeProviderNum *payeeProvider > providerNum*	8	O	O	O	O	AN	▪ Must conform to the Medicare provider check digit routine [9201] ▪ Cannot be the same stem as ServicingProviderNum [2017]
Second level Data Elements (BBeVoucher)							
VoucherId *voucherId*	2	M	M	M	M	AN	▪ Valid value: 01 [9201] ▪ If not set [9326] ▪ If greater than max vouchers (1) [2025]
ReferralIssueDate *referral > dateOfIssue*	8	-	C	-	-	D	▪ Condition**:** Cannot be set if ReferralOverrideTypeCde is set [2030] ▪ Condition: Must be set if ReferringProviderNum, ReferralPeriodTypeCde or ReferralPeriod are set. [2030] ▪ Format [9201] ▪ Must be a valid date [9201] ▪ Cannot be after the DateOfService [9309] ▪ Cannot be a future date [9204] ▪ If not set [9309]

<table>
<tr><th rowspan="2">Element Name</th><th rowspan="2">Size</th><th colspan="4">M/C/O</th><th rowspan="2">Type</th><th rowspan="2">Rule/Validation</th></tr>
<tr><th>General</th><th>Specialist</th><th>Pathology</th><th>D.Imaging</th></tr>
<tr><td>ReferralPeriodTypeCde
referral > periodTypeCde</td><td>1</td><td>-</td><td>C</td><td>-</td><td>-</td><td>A</td><td>▪ <u>Condition:</u> Cannot be set if ReferralOverrideTypeCde is set [2030]
▪ <u>Condition:</u> Must be set if ReferringProviderNum and ReferralIssueDate are set [2030]
Valid values:
▪ S = Standard
▪ N = Non standard
▪ I = Indefinite [9201]
If not set [2038]</td></tr>
<tr><td>ReferringProviderNum
referral > provider >providerNum</td><td>8</td><td>-</td><td>C</td><td>-</td><td>-</td><td>AN</td><td>▪ <u>Condition:</u> Cannot be set if ReferralOverrideTypeCde is set [2030]
▪ <u>Condition:</u> Must be set if either ReferralIssueDate, ReferralPeriod or ReferralPeriodTypeCde are present [2030]
▪ Must conform to the Medicare provider check digit routine [9201]
▪ Cannot be the same stem as ServicingProviderNum [9316]
▪ If not set [9308]</td></tr>
<tr><td>ReferralOverrideTypeCde
referralOverrideTypeCde</td><td>1</td><td>-</td><td>O</td><td>-</td><td>-</td><td>A</td><td>Valid values:
▪ L = Lost
▪ E = Emergency
▪ N = Not required [9201]
Mutually exclusive to:
▪ ReferringProviderNum
▪ ReferralPeriodTypeCde
▪ ReferralIssueDate
▪ RequestIssueDate
▪ RequestTypeCde
▪ RequestingProviderNum
▪ RequestOverrideTypeCode [2030]</td></tr>
<tr><td>RequestIssueDate
request > dateOfIssue</td><td>8</td><td>-</td><td>-</td><td>C</td><td>C</td><td>D</td><td>▪ <u>Condition:</u> Cannot be set if RequestOverrideTypeCde is set [2030]
▪ <u>Condition:</u> Must be set if RequestingProviderNum, or RequestTypeCde are set [2030]
▪ Format DDMMYYYY [9201]
▪ Must be a valid date [9201]
▪ Cannot be after the DateOfService [9311]
▪ If not set [9311]</td></tr>
<tr><td>RequestTypeCde
request > typeCde</td><td>1</td><td>-</td><td>-</td><td>C</td><td>C</td><td>A</td><td>▪ <u>Condition:</u> Cannot be set if RequestOverrideTypeCde is set [2030]
▪ <u>Condition:</u> Must be set if RequestIssueDate and RequestingProviderNum have been set [2030]
Valid values:
▪ P = Pathology
▪ D = Diagnostic Imaging[9201]
If not set [2030]</td></tr>
</table>

Element Name	Size	M/C/O General	M/C/O Specialist	M/C/O Pathology	M/C/O D.Imaging	Type	Rule/Validation
RequestingProviderNum *request > provider > providerNum*	8	-	-	C	C	AN	▪ Condition**:** Cannot be set if RequestOverrideTypeCde is set [2030] ▪ Condition**:** Must be set if RequestTypeCde or RequestIssueDate have been set [2030] ▪ Cannot be the same stem as ServicingProviderNum [9316] ▪ Must conform to the Medicare provider check digit routine [9201]

RequestOverrideTypeCode
requestOverrideTypeCde
	1	-	-	O	O	A	Valid values:
							▪ **L** = Lost
							▪ **E** = Emergency
							▪ Invalid format/value [9201]
							▪ Mutually exclusive to
							  ▪ RequestingProviderNum
							  ▪ RequestTypeCde
							  ▪ RequestIssueDate
							  ▪ [2030]

ServiceTypeCde *ServiceTypeCde*	1	M	M	M	M	A	Valid values ▪ **O** = General Practitioner ▪ **S** = Specialist ▪ **D** = Diagnostic Imaging ▪ **P =** Pathology Invalid format/value [9201] When not set [9325] If ServiceTypeCde = O, then NONE of the following are allowed: [2030] - Referral details - ReferralOverrideTypeCde - Request details - RequestOverrideTypeCde If ServiceTypeCde = S, then exactly ONE of the following must be present: [2030] - Referral details - ReferralOverrideTypeCde If ServiceTypeCde = D, then exactly ONE of the following must be present: [2030] - Request details - RequestOverrideTypeCde - SelfDeemedCde = SD - (SelfDeemedCde = SS AND Request) - (SelfDeemedCde = SS AND RequestOverrideTypeCde) If ServiceTypeCde = P, then exactly ONE of the following must be present: [2030] - Request details - RequestOverrideTypeCde - SelfDeemedCde = SD

Element Name	Size	M/C/O General	M/C/O Specialist	M/C/O Pathology	M/C/O D.Imaging	Type	Rule/Validation
Service Data Elements (BBeService)							
ServiceId *serviceId*	4	M	M	M	M	AN	▪ Valid values: 0001-0014 [9201] ▪ If not set [9332] ▪ Greater than max services [2032]
DateOfService *dateOfService*	8	M	M	M	M	D	▪ Format DDMMYYYY [9201] ▪ Cannot be a date in the future [9204] ▪ Cannot be more than 2 years in the past [9210] ▪ If not set [9306]
ItemNum *mbsItemNum*	6	M	M	M	M	AN	▪ ItemNum over 6 characters in length will be required to be rejected.[9201] ▪ If not set [9307] ▪ **Note**: If ItemNum is 6 characters, only characters 2 to 6 will be sent to the back end
ItemOverrideCde *itemOverrideCde*	2	O	O	O	O	AN	Valid values: ▪ **AP** = Not duplicate service ▪ **AO** = Not normal aftercare Invalid format/value [9201] ▪ **Note:** NC removed as a valid value in R2

RestrictiveOverrideCde
restrictiveOverrideCde
	2	O	O	-	O	AN	▪ Invalid value for Restrictive Override [9142]
							▪ Valid values:
							  ▪ **SP** = Separate sites
							  ▪ **NR** = Not related
							  ▪ **NC** = Not for comparison
							Mutually exclusive to:
							▪ ServiceTypeCde of P (2030)

LSPNum *lspnNum*	6	O	O	-	M	N	Condition: ▪ Cannot be set to 000000 [9201] ▪ If not set when [2030]
EquipmentIdNum *equipmentIdNum*	5	-	-	-	O	AN	▪ If set, LSPNNum must be set [2030] ▪ Cannot be set to 0 [9201] Mutually exclusive to: ▪ ReferralPeriodTypeCde ▪ ReferralPeriod ▪ ReferringProviderNum ▪ ReferralIssueDate ▪ [or 2030 is returned] ▪ **Note**: Only required for Radiation Oncology
SCPIdNum *scpIdNum*	4	-	-	M	-	AN	▪ Specimen Collection Point – CANNOT be 0000 [9201] ▪ Mutually exclusive to Referral elements[2030] ▪ If not set [2030]

<table>
<tr><th rowspan="2">Element Name</th><th rowspan="2">Size</th><th colspan="4">M/C/O</th><th rowspan="2">Type</th><th rowspan="2">Rule/Validation</th></tr>
<tr><th>General</th><th>Specialist</th><th>Pathology</th><th>D.Imaging</th></tr>
<tr><td>SelfDeemedCde
selfDeemedCde</td><td>2</td><td>-</td><td>-</td><td>O</td><td>O</td><td>A</td><td>Valid values:
▪ SD = Self Deemed
▪ SS = Substituted Service
▪ N = Neither Self Deemed or Substituted [9201]
  ▪ If set to SS, Request details must also be supplied [2030]
  ▪ Cannot be set to SD if the containing voucher details a request [2030]
  ▪ Cannot be set to SS if the Service Type Code = P [2030]
Mutually exclusive to:
▪ ReferralOverrideTypeCde
▪ RequestOverrideTypeCde
  ▪ [or 9207 is returned]
▪ ReferralIssueDate
▪ ReferralPeriodTypeCde
▪ ReferringProviderNum
  ▪ [or 2030 is returned]</td></tr>
</table>

Table 6. BulkBilleClaimingRequestSH

B.2.2. BulkBilleClaimingConfirmTypeRequestSH Data Elements

In Table 7. BulkBilleClaimingConfirmTypeRequestSH below the **Bold** text lists the **Element Name** and the *Italicized* text represents the *Schema Path*. In the M/C/O column **M = M**andatory, **C = C**onditional, and **O = O**ptional, while in the Type column **A**=**A**lpha, **N**=**N**umeric, **D**=**D**ate, **AN**=**A**lpha**N**umeric.

Element Name	Size	M/C/O	Type	Notes, Format, Values, Constraints
Transmission Data Elements as per MeC Technical Specification				
ConfirmType Data Elements				
ConfirmTypeInd *confirmTypeInd*	1	M	AN	Valid values: ▪ **Y** = Yes ▪ **N** = No
BenefitAssignedAuthorisedInd *benefitAssignedAuthorisedInd*	1	M	AN	Valid values: ▪ **Y** = Yes ▪ **N** = No

Table 7. BulkBilleClaimingConfirmTypeRequestSH

B.2.3. BulkBilleClaimingResponseHS Data Elements

In the Table 8. BulkBilleClaimingResponseHS below the **Bold** text lists the **`Element Name`** and the *Italicized* text represents the *`Schema Path`*. In the M/C/O column **M = M**andatory, **C = C**onditional, and **O = O**ptional, while in the Type column **A**=**A**lpha, **N**=**N**umeric, **D**=**D**ate, **AN**=**A**lpha**N**umeric.

Element Name	Size	M/C/O	Type
ClaimResponse Level			
ConcessionStatus *concessionStatus*	1	C	A
MedicareEligibilityStatus *medicareEligibilityStatus*	1	M	A
PatientMedicareCardNum *membershipReponse > currentMedicareCardNum*	10	M	N
PatientIRN *membershipReponse > currentSubnumerate*	1	M	N
PatientFirstName *membershipReponse > patient > firstName*	40	M	AN
PatientLastName *membershipReponse > patient > lastName*	40	M	AN
VoucherResponse Level			
VoucherId *voucherId*	2	M	AN
ProviderNum *providerResponse > provider > providerNum*	8	M	AN
ProviderName *providerResponse > name*	27	M	ANS
ReferringProviderNum or **RequestingProviderNum** *requestResponse/referralResponse > providerResponse >providerNum*	8	O	AN
ReferringProviderName or **RequestingProviderName** *referralResponse /requestResponse > provider > name*	40	O	ANS
ReferralIssueDate or **RequestIssueDate** *referralResponse > dateOfIssue*	8	O	D
ReferralOverrideTypeCde or **RequestOverrideTypeCde** *referralResponse /requestResponse > referral/requestOverrideTypeCde*	1	O	A
ReferralPeriodTypeCde *referralResponse > referralPeriodTypeCde*	1	O	A

Element Name	Size	M/C/O	Type
RequestTypeCde *requestResponse > requestTypeCde*	1	O	A
ServiceResponse Level Data Elements			
ServiceId *serviceId*	4	M	AN
MBSItemNumber *mbsItemNum*	6	M	AN
DateofService *dateOfService*	8	M	D
SelfDeemedCde *selfDeemedCde*	2	C	AN
LSPNNum *lspnNum*	6	O	N
EquipmentIdentificationNum *equipmentIdNum*	5	O	ANS
SCPIdentificationNum *scpIdNum*	4	O	AN
BenefitAssigned *benefitAssigned*	7	M	N

Table 8. BulkBilleClaimingResponseHS

Appendix C. Sample XML Messages

C.1. Patient Claiming (PCe) XML Messages

C.1.1. Sample PatienteClaimingRequestSH - General Practitioner Services XML Message

The following is a sample XML message containing a General Practitioner Services Patient Claim (PCe) Request:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:patienteClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
accountPaidInd="Y"><voucher serviceTypeCde="O" voucherId="01"><service
serviceId="0001" chargeAmount="5000" dateOfService="2008-03-19+11:00"
mbsItemNum="23"></service></voucher><claimant memberNum="2296753261"
memberRefNum="1"></claimant><patient memberNum="4135735042"
memberRefNum="1"></patient><servicingProvider
providerNum="2318161B"></servicingProvider></claim></ns1:patienteClaiming
Request>
```


C.1.2. Sample PatienteClaimingRequestSH - Specialist Services Request XML Message

The following is a sample XML message containing a Specialist Services Patient Claim (PCe) Request:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:patienteClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
accountPaidInd="Y"><voucher serviceTypeCde="S" voucherId="01"><service
serviceId="0001" chargeAmount="10000" dateOfService="2008-03-19+11:00"
mbsItemNum="105"></service><referral periodTypeCde="S" dateOfIssue="2008-
03-19+11:00"><provider
providerNum="2109501F"></provider></referral></voucher><claimant
memberNum="2296753441" memberRefNum="1"></claimant><patient
memberNum="4135727502" memberRefNum="1"></patient><servicingProvider
providerNum="2318161B"></servicingProvider></claim></ns1:patienteClaiming
Request>
```

C.1.3. Sample PatienteClaimingRequestSH - Diagnostic Imaging Services Request XML Message

The following is a sample XML message containing a Diagnostic Imaging Services Patient Claim (PCe) Request:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:patienteClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
accountPaidInd="Y"><voucher serviceTypeCde="D" voucherId="01"><service
serviceId="0001" chargeAmount="4000" lspnNum="1" dateOfService="2008-03-
19+11:00" mbsItemNum="57041"></service><request dateOfIssue="2008-03-
19+11:00" typeCde="D"><provider
providerNum="2074071T"></provider></request></voucher><claimant
memberNum="6501222391" memberRefNum="1"></claimant><patient
memberNum="6501222391" memberRefNum="1"></patient><servicingProvider
providerNum="2318161B"></servicingProvider></claim></ns1:patienteClaiming
Request>
```


C.1.4. Sample PatienteClaimingRequestSH - Not for Comparison - Restrictive override code NC XML Message

The following is a sample XML message containing a Patient Claim (PCe) Request:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:transmissionMessage contentsSigned="false"
contentsType="MCA/eClaiming/PatienteClaimingRequestSH@2"
transmissionId="01008032615391519040701S"
transmissionGmt="20080326153915" to="HIC00001" mode="P" from="010"
xmlns:ns1="http://hic.gov.au/hiconline/protocol/model"><adaptor
platform="os: Windows XP x86 5.1 jre: Sun 1.4.2_08"
vendor="http://hic.gov.au/" majorVersion="6.10" product="server"
minorVersion="15"></adaptor><logic platform="os: Windows XP x86 5.1 jre:
Sun 1.4.2_08" vendor="MCA" majorVersion="2" product="eClaiming"
minorVersion="9"></logic><content sendCount="1" requestId="1"
role="REQUEST" transactionId="01008032615391532345001S"
type="MCA/eClaiming/PatienteClaimingRequestSH@2" dataRef="1"
signed="false"></content><partData>
<ns1:patienteClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
accountPaidInd="Y"><voucher serviceTypeCde="D" voucherId="01"><service
restrictiveOverrideCde="NC" serviceId="0001" chargeAmount="6500"
lspnNum="1" dateOfService="2008-03-26+11:00"
mbsItemNum="57515"></service><service restrictiveOverrideCde="NC"
serviceId="0002" chargeAmount="6500" lspnNum="1" dateOfService="2008-03-
26+11:00" mbsItemNum="57515"></service><request dateOfIssue="2008-03-
20+11:00" typeCde="D"><provider
providerNum="2074071T"></provider></request></voucher><patient
memberNum="2300750671" memberRefNum="4"></patient><servicingProvider
providerNum="2109501F"></servicingProvider></claim></ns1:patienteClaiming
Request>
```

C.1.5. Sample PatienteClaimingRequestSH - Separate Site - Restrictive override code SP XML Message

The following is a sample XML message containing a Patient Claim (PCe) Request:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:transmissionMessage contentsSigned="false"
contentsType="MCA/eClaiming/PatienteClaimingRequestSH@2"
transmissionId="0100803261611232319250 1S"
transmissionGmt="20080326161123" to="HIC00001" mode="P" from="010"
xmlns:ns1="http://hic.gov.au/hiconline/protocol/model"><adaptor
platform="os: Windows XP x86 5.1 jre: Sun 1.4.2_08"
vendor="http://hic.gov.au/" majorVersion="6.10" product="server"
minorVersion="15"></adaptor><logic platform="os: Windows XP x86 5.1 jre:
Sun 1.4.2_08" vendor="MCA" majorVersion="2" product="eClaiming"
minorVersion="9"></logic><content sendCount="1" requestId="1"
role="REQUEST" transactionId="01008032616112318808101S"
type="MCA/eClaiming/PatienteClaimingRequestSH@2" dataRef="1"
signed="false"></content><partData>
<ns1:patienteClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
accountPaidInd="Y"><voucher serviceTypeCde="O" voucherId="01"><service
restrictiveOverrideCde="SP" serviceId="0001" chargeAmount="3200"
dateOfService="2008-03-26+11:00" mbsItemNum="30061"></service><service
restrictiveOverrideCde="SP" serviceId="0002" chargeAmount="3200"
dateOfService="2008-03-26+11:00" mbsItemNum="30061"></service><service
restrictiveOverrideCde="SP" serviceId="0003" chargeAmount="6075"
dateOfService="2008-03-26+11:00" mbsItemNum="30071"></service><service
restrictiveOverrideCde="SP" serviceId="0004" chargeAmount="4100"
dateOfService="2008-03-26+11:00"
mbsItemNum="30192"></service></voucher><patient memberNum="2300750671"
memberRefNum="4"></patient><servicingProvider
providerNum="2109501F"></servicingProvider></claim></ns1:patienteClaiming
Request>
```


C.1.6. Sample PatienteClaimingRequestSH - Not Related Care Plan and Consultation - Restrictive override code NR XML Message

The following is a sample XML message containing a Patient Claim (PCe) Request:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:transmissionMessage contentsSigned="false"
contentsType="MCA/eClaiming/PatienteClaimingRequestSH@2"
transmissionId="01008032616375503779901S"
transmissionGmt="20080326163755" to="HIC00001" mode="P" from="010"
xmlns:ns1="http://hic.gov.au/hiconline/protocol/model"><adaptor
platform="os: Windows XP x86 5.1 jre: Sun 1.4.2_08"
vendor="http://hic.gov.au/" majorVersion="6.10" product="server"
minorVersion="15"></adaptor><logic platform="os: Windows XP x86 5.1 jre:
Sun 1.4.2_08" vendor="MCA" majorVersion="2" product="eClaiming"
minorVersion="9"></logic><content sendCount="1" requestId="1"
role="REQUEST" transactionId="01008032616375555612701S"
type="MCA/eClaiming/PatienteClaimingRequestSH@2" dataRef="1"
signed="false"></content><partData>
<ns1:patienteClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
accountPaidInd="Y"><voucher serviceTypeCde="O" voucherId="01"><service
restrictiveOverrideCde="NR" serviceId="0001" chargeAmount="15000"
dateOfService="2008-03-26+11:00" mbsItemNum="721"></service><service
restrictiveOverrideCde="NR" serviceId="0002" chargeAmount="3000"
dateOfService="2008-03-26+11:00"
mbsItemNum="53"></service></voucher><patient memberNum="2300750671"
memberRefNum="4"></patient><servicingProvider
providerNum="0272172H"></servicingProvider></claim></ns1:patienteClaiming
Request>
```

C.1.7. Sample PatienteClaimingResponseHS Response XML Message ExplanationCde (Length 3)

The following is a sample XML message containing a PatienteClaimingResponseHS Response:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:transmissionMessage from="HIC00001" mode="P" contentsSigned="false"
contentsType="MCA/eClaiming/PatienteClaimingResponseHS@2"
transmissionGmt="20080326043915"
transmissionId="01008032615391519040701S"
xmlns:ns1="http://hic.gov.au/hiconline/protocol/model"><adaptor
vendor="http://hic.gov.au/" minorVersion="19" platform="os: AIX ppc 5.3
jre: IBM 1.4.2" product="server" majorVersion="6.10"></adaptor><logic
vendor="MCA" minorVersion="9" platform="os: AIX ppc 5.3 jre: IBM 1.4.2"
product="eClaiming" majorVersion="2"></logic><content role="RESPONSE"
statusCode="0" transactionId="0100803261539153234500 1S" sendCount="1"
requestId="1" type="MCA/eClaiming/PatienteClaimingResponseHS@2"
signed="false" dataRef="1"></content><partData>
<ns1:patienteClaimingResponse
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
assessmentStatus="OK" medicareAcceptanceType="PAID"
accountPaidInd="Y"><voucher voucherId="01"><service mbsItemNum="57515"
benefitAmount="0004590" assessmentStatus="OK" dateOfService="2008-03-
26+11:00" chargeAmount="6500" serviceId="0001"
scheduleFee="0005400"><assessmentExplanation
code="0"></assessmentExplanation></service><service mbsItemNum="57515"
benefitAmount="0004165" assessmentStatus="OK" dateOfService="2008-03-
26+11:00" chargeAmount="6500" serviceId="0002"
scheduleFee="0004900"><assessmentExplanation text="Diagnostic Imaging
Multiple Service Rule applied"
code="154"></assessmentExplanation></service></voucher><claimant
currentSubnumerate="4" currentMedicareCardNum="2300750671"><identity
lastName="VAN" firstName="FOUR"></identity></claimant><patient
currentSubnumerate="4" currentMedicareCardNum="2300750671"><identity
lastName="VAN" firstName="FOUR"></identity></patient><provider
providerNum="2109501F"><name>DR K
RANDSFORD</name></provider><assessmentError
code="0"></assessmentError></claim></ns1:patienteClaimingResponse>
```


C.1.8. Sample PatienteClaimingResponseHS Response XML Message ExplanationCde (Length 4)

The following is a sample XML message containing a PatienteClaimingResponseHS Response:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:transmissionMessage from="HIC00001" mode="P" contentsSigned="false"
contentsType="MCA/eClaiming/PatienteClaimingResponseHS@2"
transmissionGmt="20080326052730"
transmissionId="0100803261627246876910 1S"
xmlns:ns1="http://hic.gov.au/hiconline/protocol/model"><adaptor
vendor="http://hic.gov.au/" minorVersion="19" platform="os: AIX ppc 5.3
jre: IBM 1.4.2" product="server" majorVersion="6.10"></adaptor><logic
vendor="MCA" minorVersion="9" platform="os: AIX ppc 5.3 jre: IBM 1.4.2"
product="eClaiming" majorVersion="2"></logic><content role="RESPONSE"
statusCode="0" transactionId="0100803261627245966460 1S" sendCount="1"
requestId="1" type="MCA/eClaiming/PatienteClaimingResponseHS@2"
signed="false" dataRef="1"></content><partData>
```

```
<ns1:patienteClaimingResponse
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
assessmentStatus="OK" medicareAcceptanceType="NAC"
accountPaidInd="Y"><voucher voucherId="01"><service mbsItemNum="721"
assessmentStatus="ERR" dateOfService="2008-03-26+11:00"
chargeAmount="15000" serviceId="0001"><assessmentExplanation
code="9635"></assessmentExplanation></service><service mbsItemNum="53"
benefitAmount="" assessmentStatus="OK" dateOfService="2008-03-26+11:00"
chargeAmount="3000" serviceId="0002"
scheduleFee=""><assessmentExplanation
code="0"></assessmentExplanation></service></voucher><claimant
currentSubnumerate="4" currentMedicareCardNum="2300750671"><identity
lastName="VAN" firstName="FOUR"></identity></claimant><patient
currentSubnumerate="4" currentMedicareCardNum="2300750671"><identity
lastName="VAN" firstName="FOUR"></identity></patient><provider
providerNum="2074071T"><name>DR B
HARPER</name></provider><assessmentError
code="0"></assessmentError></claim></ns1:patienteClaimingResponse>
```

C.1.9. Sample PatienteClaimingCancelRequestSH Cancel Request XML Message

The following is a sample XML message containing a PatienteClaimingCancelRequestSH Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:patienteClaimingCancelRequest acceptInd="N"
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-
2"></ns1:patienteClaimingCancelRequest>
```

C.1.10. Sample PatienteClaimingCancelResponseHS Cancel Response XML Message

The following is a sample XML message containing a PatienteClaimingCancelResponseHS Cancel Response, which has no body:

```
<content statusCode="0" requestId="1"
transactionId="01007030910081411898391M" sendCount="1" role="RESPONSE"
dataRef="0" type="MCA/eClaiming/PatienteClaimingCancelResponseHS@2"
signed="false"></content>
```

C.2. Bulk Billing (BBe) XML Messages

C.2.1. Sample Bulk Billing (BBe) General Practitioner Services Request XML Message

The following is a sample XML message containing a Bulk Billing (BBe) General Practitioner Services Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:bulkBilleClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
cevRequestInd="Y"><voucher serviceTypeCde="O" voucherId="01"><service
mbsItemNum="23" dateOfService="2008-03-19+11:00"
serviceId="0001"></service></voucher><patient memberNum="4134636471"
memberRefNum="1"></patient><servicingProvider
providerNum="2392371A"></servicingProvider></claim></ns1:bulkBilleClaimin
gRequest>
```

C.2.2. Sample Bulk Billing (BBe) Specialist Services Request XML Message

The following is a sample XML message containing a Bulk Billing (BBe) Specialist Services Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<ns1:bulkBilleClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-
2"><claim><voucher serviceTypeCde="S" referralOverrideTypeCde="L"
voucherId="01"><service mbsItemNum="105" dateOfService="2008-03-20+11:00"
serviceId="0001"></service></voucher><patient memberNum="4134636471"
memberRefNum="1"></patient><servicingProvider
providerNum="2074071T"></servicingProvider></claim></ns1:bulkBilleClaimin
gRequest>
```


C.2.3. Sample Bulk Billing (BBe) Pathology Services Request XML Message

The following is a sample XML message containing a Bulk Billing (BBe) Pathology Services Request:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:bulkBilleClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
cevRequestInd="Y"><voucher serviceTypeCde="P" voucherId="01"><service
scpIdNum="0001" mbsItemNum="65120" dateOfService="2008-03-20+11:00"
serviceId="0001"></service><request dateOfIssue="2007-01-11+11:00"
typeCde="P"><provider
providerNum="2109501F"></provider></request></voucher><patient
memberNum="2300178451" memberRefNum="1"></patient><servicingProvider
providerNum="2395321J"></servicingProvider></claim></ns1:bulkBilleClaimin
gRequest>
```

C.2.4. Sample Bulk Billing (BBe) Diagnostic Imaging Services Request XML Message

The following is a sample XML message containing a Bulk Billing (BBe) Diagnostic Imaging Services Request:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:bulkBilleClaimingRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
cevRequestInd="Y"><voucher serviceTypeCde="D" voucherId="01"><service
mbsItemNum="57506" lspnNum="1" dateOfService="2008-03-20+11:00"
serviceId="0001"></service><service mbsItemNum="58100" lspnNum="1"
dateOfService="2008-03-20+11:00" serviceId="0002"></service><request
dateOfIssue="2008-03-19+11:00" typeCde="D"><provider
providerNum="2074071T"></provider></request></voucher><patient
memberNum="2300178451" memberRefNum="1"></patient><servicingProvider
providerNum="2318161B"></servicingProvider></claim></ns1:bulkBilleClaimin
gRequest>
```


C.2.5. Sample BulkBilleClaimingConfirmTypeRequestSH Confirm Type Request XML Message

The following is a sample XML message containing a BulkBilleClaimingConfirmTypeRequestSH Confirm Type Request:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:BulkBilleClaimingConfirmRequest
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-
2"><confirm><confirmTypeInd>Y</confirmTypeInd><benefitAssignmentAuthorise
dInd>Y</benefitAssignmentAuthorisedInd></confirm></ns1:BulkBilleClaimingC
onfirmRequest>
```


C.2.6. Sample BulkBilleClaimingResponseHS Response XML Message

This response covers General Practitioner Services, Specialist Services, Pathology Services, and Diagnostic Imaging Services.

The following is a sample XML message containing a BulkBilleClaimingResponseHS Response:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:bulkBilleClaimingResponse
xmlns:ns1="http://medicareaustralia.gov.au/eclaiming/version-2"><claim
concessionStatus="Y" medicareEligibilityStatus="Y"><voucher
voucherId="01"><service serviceId="0001" benefitAssigned="2975"
dateOfService="2008-03-20+11:00" lspnNum="1"
mbsItemNum="57506"></service><service serviceId="0002"
benefitAssigned="2975" dateOfService="2008-03-20+11:00" lspnNum="1"
mbsItemNum="57506"></service><request dateOfIssue="2008-03-19+11:00"
typeCde="D"><provider providerNum="2074071T"><name>DR BEN
HARPER</name></provider></request><provider
providerNum="2318161B"><name>DR MEAGAN
CARDWELL</name></provider></voucher><patient currentSubnumerate="1"
currentMedicareCardNum="2300178451"><identity lastName="YIPPETT"
firstName="CORNELIUS"></identity></patient></claim></ns1:bulkBilleClaimin
gResponse>
```

C.2.7. Sample BulkBilleClaimingConfirmTypeResponseHS Confirm Type Response XML Message

The following is a sample XML message containing a BulkBilleClaimingConfirmTypeResponseHS Confirm Type Response (Status code 1716 – successful, 0 – deleted):

```
<content requestId="1" transactionId="0100801251445535330 2101M"
statusCode="1716" sendCount="1" role="REQUEST_ACK"
type="MCA/eClaiming/BulkBilleClaimingConfirmTypeResponseHS@2" dataRef="0"
signed="false"></content>
```

Appendix D. Provider and Card Validation Routines

The following validation routines are used by the Server Adaptor to verify the correct keying of some manually entered data:

- Medicare Card Number Validation
- Provider Number Validation
- Ancillary Provider Number Validation

D.1. Medicare Card Number Validation

The following algorithm is the Medicare Check Digit Routine:

- (digit 1) +
- (digit 2 * 3) +
- (digit 3 * 7) +
- (digit 4 * 9) +
- (digit 5) +
- (digit 6 * 3) +
- (digit 7 * 7) +
- (digit 8 * 9)
- divide the total by 10

The remainder is the Check Digit. The check digit becomes the ninth digit in the Medicare card number.

D.2. Provider Number Validation

The Provider Number is comprised of:

- Provider Stem - a six-digit number.
- 1 Practice Location Character (PLV) - see below
- 1 Check Digit

The algorithm used for the Provider Check Digit is:

- (digit 1 * 3) +
- (digit 2 * 5) +
- (digit 3 * 8) +
- (digit 4 * 4) +
- (digit 5 * 2) +
- (digit 6) +
- (PLV * 6)
- divide the result by 11

The remainder is allocated an alpha that is the provider check digit (refer to the table following for details).

D.2.1. Practice Location Character (PLV)

The Practice Location Character indicates the order of the practices that a provider has been registered at. Each Practice Location Character is allocated a Practice Location Value (PLV) that is used in the Provider Check Digit Routine.

Character	Value	Character	Value
0	0	G	16
1	1	H	17
2	2	J	18
3	3	K	19
4	4	L	20
5	5	M	21
6	6	N	22
7	7	P	23
8	8	Q	24
9	9	R	25
A	10	T	26
B	11	U	27
C	12	V	28
D	13	W	29
E	14	X	30
F	15	Y	31

Table 9. Practice Location Character

D.2.2. Check Digit

The Check Digits for the resultant remainder are as follows:

Remainder	Check-digit
0	Y
1	X
2	W
3	T
4	L
5	K
6	J
7	H
8	F
9	B
10	A

Table 10. Check Digit

Appendix E. Message Headers

The handlers configured in the Server Adaptor's quick-post service will then deal with this message based on the incoming HTTP headers in order to on-send the message to Medicare Australia.

Most of the headers are self-explanatory, but the following are of particular importance. The two tables in Appendix E.1 Request Messages Headers below and Appendix E.2 Response Message Headers, below detail the components of the headers based on whether they are Request or Response related.

E.1. Request Messages Headers

Quick Post Protocol Header Name	Example Header Value	Purpose
hiconline.protocol.content.transactionid	01008010711445733156101M	▪ Specifies the unique identifier for this message content – that is, the Transaction ID.
hiconline.protocol.content.role	REQUEST	▪ Specifies the role of this content part.
hiconline.protocol.remote.contenttype	MCA/eClaiming/PatienteClaimingRequestSH@2	▪ Specifies the contentType for this content part and transmission.
hiconline.protocol.remote.mode	P	▪ Signifies production, development or testing modes. ▪ Should normally be set to P.

When sending outbound messages, it is not mandatory to supply all headers listed in the table above. The logic pack either has default values for some of these, or generates the values as required. The mandatory headers are:

- hiconline.protocol.content.transactionid
- hiconline.protocol.remote.contenttype

XML schemas provided with the Medicare Easyclaim Server Adaptor documentation, and in Appendix F Medicare Easyclaim Schema on page 50 and Appendix G HICOnline Support Schema on page 56, will be needed in order to generate the content payload in the body of the POST. Bindings can be created against these using the tools of choice.

E.2. Response Message Headers

Quick Post Content Header Name	Example Header Value	Purpose
hiconline.protocol.content.transactionid	01008010711415998005901M	▪ Specifies the unique identifier for this message content – that is, the Transaction ID.
hiconline.protocol.content.role	RESPONSE	▪ Specifies the role of this content part.
hiconline.protocol.contenttype	MCA/eClaiming/PatienteClaimingResponseHS@2	▪ Specifies the contentType for this content part and transmission.
hiconline.protocol.remote.statuscode	0	▪ Specifies the statusCode for this content part or transmission – if response.
hiconline.protocol.response.msgOrigincode	MedicareAustralia	▪ Indicates that the response is coming from Medicare Australia's hub, and that a transaction fee is payable.

E.3. TransactionId

In Table 11. TransactionId validation below the **Bold** text lists the **`Element Name`** and the *Italicized* text represents the `Schema Path`. In the M/C/O column **M = M**andatory, **C = C**onditional, and **O = O**ptional, while in the Type column **A**=**A**lpha, **N**=**N**umeric, **D**=**D**ate, **AN**=**A**lpha**N**umeric.

Element Name	Size	M/C/O	Type	Notes, Format, Values, Constraints

`TransactionId`	24	M	AN	Format:
				BBBYYMMDDHHMMSS123456%#+
				Values:
				▪ BBB = 3 DIGIT PARTICIPANT Financial Institution Id as per APCA Standards
				▪ YYMMDD = Date (using AEST or AEDST)
				▪ HHMMSS = Time (using AEST or AEDST)
				▪ 123456 = 6 character unique numerical Id
				▪ %
				  ▪ For PCe,
				  0 = will be for an initial claim, and where a PCe Claim transaction is to be cancelled, the % will be 9
				  ▪ For BBE,
				  0 = will be for an initial claim, 2 = will be where the claim transaction is accepted, and where the BBe Claim transaction is to be declined, the % will be 9
				▪ # = 1 (Patient Claim) or, **2** (Bulk Bill Claim)
				▪ **+** = M (Manual entry) or, **S** (Swipe card)

Table 11. TransactionId validation

Appendix F. Medicare Easyclaim Schema

F.1. Medicare Easyclaim Schema


```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    ! ***************************************************************************************
    ! NOTE:
    !       1.      This schema was created on 10/10/06 for the
    !               eClaiming Medicare Easyclaim project.
    !   2. Schema was updated on 10/11/2006 to reflect changes in business requirements Version 1.9.
    !   3. Schema was updated on 21/02/2006 to reflect changes in business requirements
    !       4. Schema was updated on 21/01/2008 to reflect changes in business requirements Version 2.0
    !   5. Schema was updated on 31/01/2008 to add serviceTypeCde for BBe

    ! ***************************************************************************************
    !-->

<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://medicareaustralia.gov.au/eclaiming/version-2"
xmlns:support="http://hic.gov.au/hiconline/support/version-4" xmlns:jxb="http://java.sun.com/xml/ns/jaxb"
jxb:version="1.0" xmlns:tns="http://medicareaustralia.gov.au/eclaiming/version-2">

    <import namespace="http://hic.gov.au/hiconline/support/version-4"
schemaLocation="HicOnlineSupport.xsd"/>

    <!--
            ! For internal use only - used in generation of Java bindings
            !-->
    <xsd:annotation>
            <xsd:appinfo>
                    <jxb:globalBindings collectionType="indexed" generateIsSetMethod="true"/>
                    <jxb:schemaBindings>
                            <jxb:package name="au.gov.medicareaustralia.biz.eclaiming.model"/>
                            <jxb:nameXmlTransform>
                                    <jxb:typeName suffix="Details"/>
                            </jxb:nameXmlTransform>
                    </jxb:schemaBindings>
            </xsd:appinfo>
    </xsd:annotation>

    <!--
            !  Root Elements for Patient eClaiming Medicare Easyclaim
            !-->
    <xsd:element name="patienteClaimingRequest" type="tns:PatienteClaimingRequest"/>
    <xsd:element name="patienteClaimingResponse" type="tns:PatienteClaimingResponse"/>

    <!--
            !  Root Elements for Bulk Bill eClaiming Medicare Easyclaim
            !-->
    <xsd:element name="bulkBilleClaimingRequest" type="tns:BulkBilleClaimingRequest"/>
    <xsd:element name="bulkBilleClaimingResponse" type="tns:BulkBilleClaimingResponse"/>

    <!--
            !  Root Elements for Patient eClaiming Medicare Easyclaim Cancel
            !-->
    <xsd:element name="patienteClaimingCancelRequest" type="tns:PatientClaimeClaimingDeclineTypeRequest"/>

    <!--
            !  Root Elements for Patient eClaiming Medicare Easyclaim Cancel
            !-->
    <xsd:element name="BulkBilleClaimingConfirmRequest" type="tns:BulkBilleClaimingConfirmTypeRequest"/>

    <!--
            !  Root Elements for Bulk Bill eClaiming Medicare Easyclaim Stored Object - medicare australia
internal use only
            !--><xsd:element name="bulkBilleClaimingStoredRequest"
type="tns:BulkBilleClaimingStoredRequest"/>

    <!--
            ! Patient eClaiming Medicare Easyclaim request
            !-->
    <xsd:complexType name="PatienteClaimingRequest">
            <xsd:sequence>
                    <xsd:element name="claim" type="tns:PCeClaim"/>
```

```
        </xsd:sequence>
</xsd:complexType>

<!--
        ! Patient eClaiming Medicare Easyclaim request - Claim level element
        !-->
<xsd:complexType name="PCeClaim">
        <xsd:sequence>
                <xsd:element name="voucher" type="tns:PCeVoucher" minOccurs="1" maxOccurs="unbounded"/>
                <xsd:element name="claimant" type="support:Membership" minOccurs="0"/>
                <xsd:element name="patient" type="support:Membership" minOccurs="0"/>
                <xsd:element name="servicingProvider" type="support:Provider" minOccurs="0"/>
                <xsd:element name="payeeProvider" type="support:Provider" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="accountPaidInd" type="support:IndicatorEnum"/>
        <xsd:attribute name="accountReferenceNum" type="string"/>
</xsd:complexType>

<!--
        ! Patient eClaiming Medicare Easyclaim request - Voucher level element
        !-->
<xsd:complexType name="PCeVoucher">
        <xsd:sequence>
                <xsd:element name="service" type="tns:PCeService" minOccurs="1" maxOccurs="unbounded"/>
                <xsd:element name="referral" type="tns:Referral" minOccurs="0"/>
                <xsd:element name="request" type="tns:Request" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="serviceTypeCde" type="tns:ServiceTypeCdeEnum"/>
        <xsd:attribute name="referralOverrideTypeCde" type="string"/>
        <xsd:attribute name="requestOverrideTypeCde" type="string"/>
        <xsd:attribute name="voucherId" type="string" use="required"/>
</xsd:complexType>

<!--
        ! Patient eClaiming Medicare Easyclaim request - Service level element
        !-->
<xsd:complexType name="PCeService">
        <xsd:attribute name="selfDeemedCde" type="tns:SelfDeemedCdeEnum"/>
        <xsd:attribute name="serviceId" type="string" use="required"/>
        <xsd:attribute name="dateOfService" type="date"/>
        <xsd:attribute name="mbsItemNum" type="string"/>
        <xsd:attribute name="chargeAmount" type="string"/>
        <xsd:attribute name="lspnNum" type="string"/>
        <xsd:attribute name="equipmentIdNum" type="string"/>
        <xsd:attribute name="patientContribAmt" type="string"/>
        <xsd:attribute name="itemOverrideCde" type="string"/>
        <xsd:attribute name="restrictiveOverrideCde" type="string"/>
</xsd:complexType>

<!--
        ! Referral complex type is the referral element attached to voucher
        !-->
<xsd:complexType name="Referral">
        <xsd:sequence>
                <xsd:element name="provider" type="support:Provider" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="dateOfIssue" type="date"/>
        <xsd:attribute name="periodTypeCde" type="tns:ReferralPeriodTypeCdeEnum"/>
</xsd:complexType>

<!--
        ! ReferralResponse complex type is the referral element attached to response voucher
        !-->
<xsd:complexType name="ReferralResponse">
        <xsd:sequence>
                <xsd:element name="provider" type="tns:ProviderResponse" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="dateOfIssue" type="date"/>
        <xsd:attribute name="periodTypeCde" type="tns:ReferralPeriodTypeCdeEnum"/>
</xsd:complexType>

<!--
        ! RequestResponse complex type is the request element attached to response voucher
        !-->
<xsd:complexType name="RequestResponse">
```

```
        <xsd:sequence>
                <xsd:element name="provider" type="tns:ProviderResponse" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="dateOfIssue" type="date"/>
        <xsd:attribute name="typeCde" type="tns:RequestTypeCdeEnum"/>
</xsd:complexType>

<!--
        ! Request complex type is the request element attached to voucher
        !-->
<xsd:complexType name="Request">
        <xsd:sequence>
                <xsd:element name="provider" type="support:Provider" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="dateOfIssue" type="date"/>
        <xsd:attribute name="typeCde" type="tns:RequestTypeCdeEnum"/>
</xsd:complexType>

<!--
        ! periodTypeCde:
        ! * S  indicates Standard
        ! * N  indicates Non Standard
        ! * I   indicates Indefinite
        !-->
<xsd:simpleType name="ReferralPeriodTypeCdeEnum">
        <xsd:restriction base="string">
                <xsd:enumeration value="S"/>
                <xsd:enumeration value="N"/>
                <xsd:enumeration value="I"/>
        </xsd:restriction>
</xsd:simpleType>

<!--
        ! typeCde:
        ! * P  indicates Pathology
        ! * D  indicates Dianostic Imaging
        !-->
<xsd:simpleType name="RequestTypeCdeEnum">
        <xsd:restriction base="string">
                <xsd:enumeration value="P"/>
                <xsd:enumeration value="D"/>
        </xsd:restriction>
</xsd:simpleType>

<!--
        ! SelfDeemedCde:
        ! * N   indicates Not Self Deemed
        ! * SD  indicates Self Determined
        ! * SS          indicates Substituted
        ! * &quot; &quot;      indicates None of the above
        !-->
<xsd:simpleType name="SelfDeemedCdeEnum">
        <xsd:restriction base="string">
                <xsd:enumeration value="N"/>
                <xsd:enumeration value="SD"/>
                <xsd:enumeration value="SS"/>
                <xsd:enumeration value=" "/>
        </xsd:restriction>
</xsd:simpleType>

<!--
        ! serviceTypeCde:
        ! * O   indicates Other (General Service Item)
        ! * P   indicates Pathology
        ! * S   indicates Specialist
        ! * D   indicates Diagnostic
        ! * E   indicates
        !-->
<xsd:simpleType name="ServiceTypeCdeEnum">
        <xsd:restriction base="string">
                <xsd:enumeration value="O"/>
                <xsd:enumeration value="P"/>
                <xsd:enumeration value="S"/>
                <xsd:enumeration value="D"/>
        </xsd:restriction>
</xsd:simpleType>
```

```
<!--
        ! Patient eClaiming Medicare Easyclaim response element
        ! Root element containing Medicare assessment response
        !-->

<xsd:complexType name="ProviderResponse">
        <xsd:complexContent>
                <xsd:extension base="support:Provider">
                        <xsd:sequence>
                                <xsd:element name="name" type="string" minOccurs="0"/>
                        </xsd:sequence>
                </xsd:extension>
        </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="PatienteClaimingResponse">
        <xsd:sequence>
                <xsd:element name="claim" type="tns:PCeClaimResponse"/>
        </xsd:sequence>
</xsd:complexType>

<!--
        ! Patient eClaiming Medicare Easyclaim response - Claim level assessment element
        !-->
<xsd:complexType name="PCeClaimResponse">
        <xsd:sequence>
                <xsd:element name="voucher" type="tns:PCeVoucherResponse" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element name="claimant" type="tns:MembershipResponse" minOccurs="0"/>
                <xsd:element name="patient" type="tns:MembershipResponse" minOccurs="0"/>
                <xsd:element name="provider" type="tns:ProviderResponse" minOccurs="0"/>
                <xsd:element name="assessmentError" type="support:Explanation" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="accountPaidInd" type="support:IndicatorEnum"/>
        <xsd:attribute name="assessmentStatus" type="string"/>
        <xsd:attribute name="medicareAcceptanceType" type="string"/>
</xsd:complexType>

<!--
        ! Patient eClaiming Medicare Easyclaim response - Voucher level element
        !-->
<xsd:complexType name="PCeVoucherResponse">
        <xsd:sequence>
                <xsd:element name="service" type="tns:PCeServiceResponse" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>

        <xsd:attribute name="voucherId" type="string" use="required"/>
</xsd:complexType>

<!--
        ! Patient eClaiming Medicare Easyclaim response - Service level element
        !-->
<xsd:complexType name="PCeServiceResponse">
        <xsd:sequence>
                <xsd:element name="assessmentExplanation" type="support:Explanation" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="serviceId" type="string" use="required"/>
        <xsd:attribute name="mbsItemNum" type="string"/>
        <xsd:attribute name="dateOfService" type="date"/>
        <xsd:attribute name="chargeAmount" type="string"/>
        <xsd:attribute name="scheduleFee" type="string"/>
        <xsd:attribute name="patientContribAmt" type="string"/>
        <xsd:attribute name="benefitAmount" type="string"/>
        <xsd:attribute name="assessmentStatus" type="string"/>
</xsd:complexType>

<xsd:complexType name="MembershipResponse">
        <xsd:sequence>
                <xsd:element name="identity" type="support:Identity" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="currentMedicareCardNum" type="string"/>
        <xsd:attribute name="currentSubnumerate" type="string"/>
        <xsd:attribute name="currentMedicareCardIssueNum" type="string"/>
</xsd:complexType>
```

```
<!-- Confirm Request Type
        * BenefitAuthorised is not required for PCe Claiming.
        !-->
<xsd:complexType name="PatientClaimeClaimingDeclineTypeRequest">
        <xsd:attribute name="acceptInd" type="string"/>
</xsd:complexType>


<!-- Bulk Bill Request
        !-->
<xsd:complexType name="BulkBilleClaimingRequest">
        <xsd:sequence>
                <xsd:element name="claim" type="tns:BBeClaim"/>
        </xsd:sequence>
</xsd:complexType>

<!-- Bulk Bill eClaiming Medicare Easyclaim Request - Claim level element
        !-->
<xsd:complexType name="BBeClaim">
        <xsd:sequence>
                <xsd:element name="voucher" type="tns:BBeVoucher" minOccurs="1" maxOccurs="unbounded"/>
                <xsd:element name="patient" type="support:Membership" minOccurs="0"/>
                <xsd:element name="servicingProvider" type="support:Provider" minOccurs="0"/>
                <xsd:element name="payeeProvider" type="support:Provider" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="cevRequestInd" type="support:IndicatorEnum"/>
</xsd:complexType>

<!-- Bulk Bill eClaiming Medicare Easyclaim Request - Voucher level element
        !-->
<xsd:complexType name="BBeVoucher">
        <xsd:sequence>
                <xsd:element name="service" type="tns:BBeService" minOccurs="1" maxOccurs="unbounded"/>
                <xsd:element name="referral" type="tns:Referral" minOccurs="0"/>
                <xsd:element name="request" type="tns:Request" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="serviceTypeCde" type="tns:ServiceTypeCdeEnum"/>
        <xsd:attribute name="voucherId" type="string" use="required"/>
        <xsd:attribute name="referralOverrideTypeCde" type="string"/>
        <xsd:attribute name="requestOverrideTypeCde" type="string"/>

</xsd:complexType>

<!-- Bulk Bill eClaiming Medicare Easyclaim Request - Service level element
        !-->
<xsd:complexType name="BBeService">
        <xsd:attribute name="selfDeemedCde" type="tns:SelfDeemedCdeEnum"/>
        <xsd:attribute name="serviceId" type="string" use="required"/>
        <xsd:attribute name="mbsItemNum" type="string"/>
        <xsd:attribute name="dateOfService" type="date"/>
        <xsd:attribute name="itemOverrideCde" type="string"/>
        <xsd:attribute name="restrictiveOverrideCde" type="string"/>
        <xsd:attribute name="lspnNum" type="string"/>
        <xsd:attribute name="equipmentIdNum" type="string"/>
        <xsd:attribute name="scpIdNum" type="string"/>

</xsd:complexType>

<!-- Bulk Bill eClaiming Medicare Easyclaim assessment response
        ! Root element containing Medicare acknowledgment response
        !-->
<xsd:complexType name="BulkBilleClaimingResponse">
        <xsd:sequence>
                <xsd:element name="claim" type="tns:BBeClaimResponse"/>
        </xsd:sequence>
</xsd:complexType>

<!-- Bulk Bill eClaiming Medicare Easyclaim Response - Claim level acknowledgment element
        !-->
<xsd:complexType name="BBeClaimResponse">
        <xsd:sequence>
                <xsd:element name="voucher" type="tns:BBeVoucherResponse" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element name="patient" type="tns:MembershipResponse" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="concessionStatus" type="string"/>
```

```
        <xsd:attribute name="medicareEligibilityStatus" type="string"/>
    </xsd:complexType>

    <!--
        ! Bulk Bill eClaiming Medicare Easyclaim response - Voucher level element
        !-->
    <xsd:complexType name="BBeVoucherResponse">
        <xsd:sequence>
            <xsd:element name="service" type="tns:BBeServiceResponse" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="referral" type="tns:ReferralResponse" minOccurs="0"/>
            <xsd:element name="request" type="tns:RequestResponse" minOccurs="0"/>
            <xsd:element name="provider" type="tns:ProviderResponse" minOccurs="0"/>
        </xsd:sequence>

        <xsd:attribute name="voucherId" type="string" use="required"/>
        <xsd:attribute name="referralOverrideTypeCde" type="string"/>
        <xsd:attribute name="requestOverrideTypeCde" type="string"/>
    </xsd:complexType>

    <!--
        ! Bulk Bill eClaiming Medicare Easyclaim response - Service level element
        !-->
    <xsd:complexType name="BBeServiceResponse">
        <xsd:attribute name="selfDeemedCde" type="tns:SelfDeemedCdeEnum"/>
        <xsd:attribute name="mbsItemNum" type="string"/>
        <xsd:attribute name="dateOfService" type="date"/>
        <xsd:attribute name="lspnNum" type="string"/>
        <xsd:attribute name="equipmentIdNum" type="string"/>
        <xsd:attribute name="serviceId" type="string" use="required"/>
        <xsd:attribute name="benefitAssigned" type="string"/>
        <xsd:attribute name="scpIdNum" type="string"/>

    </xsd:complexType>

    <!-- Confirm Request Type
        !-->
    <xsd:complexType name="BulkBilleClaimingConfirmTypeRequest">
        <xsd:sequence>
            <xsd:element name="confirm" type="tns:ConfirmTypeRequest" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>

    <!-- Confirm Request Type- Top level element
        !-->
    <xsd:complexType name="ConfirmTypeRequest">
        <xsd:sequence>
            <xsd:element name="confirmTypeInd" type="support:IndicatorEnum"/>
            <xsd:element name="benefitAssignmentAuthorisedInd" type="support:IndicatorEnum"/>
        </xsd:sequence>
    </xsd:complexType>

    <!-- BulkBilleClaimingStoredObject - medicare australia internal use only
        !-->

  <xsd:complexType name="BulkBilleClaimingStoredRequest">
        <xsd:sequence>
            <xsd:element name="claim" type="tns:BBeClaim"/>
            <xsd:element name="benefit" type="tns:BenefitAmount" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="BenefitAmount">
        <xsd:attribute name="amount" type="string" use="required"/>
    </xsd:complexType>

</xsd:schema>
```

Appendix G. HICOnline Support Schema

G.1. HICOnline Support Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--
	!
*********************************************************************
	! NOTE:
	!	1.	This schema (Support) was last modified on 20/08/04
	!		for adding Release 4 elements.
	!
	!	2.	Modified on 01/04/2005 for adding Release 5 - OEC
	!		elements.
	!
	!	3.	Modified on 23/01/2006 for adding Release 6.
	!
*********************************************************************
	-->

<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://hic.gov.au/hiconline/support/version-4"
xmlns:tns="http://hic.gov.au/hiconline/support/version-4"
xmlns:jxb="http://java.sun.com/xml/ns/jaxb" jxb:version="1.0">

	<!--
		! For internal use only - used in generation of Java
bindings
		! Global bindings used to generate this project are in
HicOnlineSupprt.xjb
		!-->
	<xsd:annotation>
		<xsd:appinfo>
		<!--<jxb:globalBindings collectionType="indexed"
generateIsSetMethod="true" />-->
			<jxb:schemaBindings>
				<jxb:package
name="au.gov.hic.hiconline.biz.support.model" />
				<jxb:nameXmlTransform>
					<jxb:typeName suffix="Details" />
				</jxb:nameXmlTransform>
			</jxb:schemaBindings>
		</xsd:appinfo>
	</xsd:annotation>


	<!--
		! gender:
		! * M  indicates Male
		! * F  indicates Female
		!-->
	<xsd:simpleType name="GenderEnum">
		<xsd:restriction base="xsd:string">
			<xsd:enumeration value="M" />
			<xsd:enumeration value="F" />
		</xsd:restriction>
	</xsd:simpleType>
```

```
      <xsd:complexType name="Person">
            <xsd:sequence>
                  <xsd:element name="identity" type="tns:Identity" />
                  <xsd:element name="alias" type="tns:Identity"
minOccurs="0" />
                  <!--Element "ResidentialAddress" is used by IMC PC
only 17/08/04-->
                  <xsd:element name="residentialAddress"
type="tns:Address" minOccurs="0" />
            </xsd:sequence>
            <xsd:attribute name="dateOfBirth" type="xsd:date" />
            <xsd:attribute name="gender" type="xsd:string" />
      </xsd:complexType>

      <xsd:complexType name="Address">
            <xsd:attribute name="addressLineOne" type="xsd:string" />
            <xsd:attribute name="addressLineTwo" type="xsd:string" />
            <xsd:attribute name="locality" type="xsd:string" />
            <!--change postcode from int to string on 09/09/04-->
            <xsd:attribute name="postcode" type="xsd:string" />
      </xsd:complexType>

      <xsd:complexType name="Identity">
            <xsd:attribute name="firstName" type="xsd:string" />
            <xsd:attribute name="lastName" type="xsd:string" />
            <xsd:attribute name="secondInitial" type="xsd:string" />
      </xsd:complexType>

   <!--
    ! Simple name + value pair
    !-->
   <xsd:complexType name="Property" >
      <xsd:attribute name="name" type="xsd:string" use="required" />
      <xsd:attribute name="value" type="xsd:string" use="required" />
   </xsd:complexType>

      <xsd:complexType name="Membership">
            <xsd:sequence>
                  <xsd:element name="attributes" type="tns:Property"
minOccurs="0" maxOccurs="unbounded" />
            </xsd:sequence>
            <xsd:attribute name="organisation" type="xsd:string" />
            <xsd:attribute name="memberNum" type="xsd:string" />
            <xsd:attribute name="memberRefNum" type="xsd:string" />
      </xsd:complexType>

      <xsd:complexType name="MembershipStatus">
            <xsd:sequence>
                  <xsd:element name="currentMembership"
type="tns:Membership" minOccurs="0" />
                  <xsd:element name="currentMember"
type="tns:Identity" minOccurs="0" />
            </xsd:sequence>
            <xsd:attribute name="statusCode" type="xsd:int" />
            <xsd:attribute name="procDate" type="xsd:date" />
      </xsd:complexType>

      <!--
            ! HicOnlineSupport.xsd was modified on 1/4/05 for OEC R5
            ! -->

      <xsd:simpleType name="OECEstimationCdeEnum">
            <xsd:restriction base="xsd:string">
```

```
                  <xsd:enumeration value="A" />
                  <xsd:enumeration value="R" />
                  <xsd:enumeration value="W" />
            </xsd:restriction>
      </xsd:simpleType>

      <xsd:complexType name="OECEstimationReturn">
            <xsd:attribute name="responseCde"
type="tns:OECEstimationCdeEnum" />
            <xsd:attribute name="explanationCode" type="xsd:int" />
            <xsd:attribute name="explanationText" type="xsd:string" />
      </xsd:complexType>

      <xsd:complexType name="PatientPayment">
            <xsd:attribute name="amount" type="xsd:int" />
            <xsd:attribute name="amountDescription" type="xsd:string"
/>
      </xsd:complexType>


      <xsd:complexType name="ExcessPayment">
            <xsd:complexContent>
                  <xsd:extension base="tns:PatientPayment">
                        <xsd:attribute name="bonusAmount"
type="xsd:int" />
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>


      <xsd:complexType name="CoPayment">
            <xsd:complexContent>
                  <xsd:extension base="tns:PatientPayment">
                        <xsd:attribute name="daysRemaining"
type="xsd:int" />
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>


      <xsd:complexType name="Accident">
            <xsd:attribute name="accidentDate" type="xsd:date" />
            <xsd:attribute name="accidentInd" type="xsd:boolean" />
      </xsd:complexType>

      <xsd:complexType name="HealthFundTable">
            <xsd:attribute name="tableName" type="xsd:string" />
            <xsd:attribute name="tableDescription" type="xsd:string"
/>
            <xsd:attribute name="tableScale" type="xsd:string" />
      </xsd:complexType>

      <!--
            ! Provider complexType contains providerNum and email as
attributes
            ! Note on Attributes:
            !  providerNum is a type string and it is use as mandatory
for Provider
            !-->
      <xsd:complexType name="Provider">
            <xsd:attribute name="providerNum" type="xsd:string"
use="required" />
      </xsd:complexType>
```

```
      <!--
            ! ProcessTimeout complexType contains timeout string for
long running requests
            ! Note on Attributes:
            !  timeout is a type int (whole hours) and is use optional
            !-->
      <xsd:complexType name="ProcessTimeout">
            <xsd:attribute name="timeout" type="xsd:int"
use="optional" />
      </xsd:complexType>

      <!--
******************************************************************* -->
      <!-- Release 6 changes.
-->
      <!--
******************************************************************* -->
      <!--
            ! Explation complexType contains code and text fields
            !
            !-->
      <xsd:complexType name="Explanation">
            <xsd:attribute name="code" type="xsd:string" />
            <xsd:attribute name="text" type="xsd:string" />
      </xsd:complexType>

       <!--
    ! Base type for those complex types that are structural parts of a
request or response.<p>
    ! -->
   <xsd:complexType name="MessagePart" abstract="true">
      <xsd:attribute name="messagePartId" type="xsd:ID"
         use="optional" />
   </xsd:complexType>


   <!--
    ! Base type for those complex types that are requests or
responses.
    ! -->
   <xsd:complexType name="Message" abstract="true" >
   </xsd:complexType>


   <!--
    ! A generic message part that just has a single text field
    !-->
   <xsd:complexType name="TextPart">
      <xsd:complexContent>
         <xsd:extension base="tns:MessagePart">
            <xsd:sequence>
               <xsd:element name="text" type="xsd:string" />
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>

  <!--
    ! Describes position of a message (or other entity) within a
sequence.
    ! F   FIRST
    ! L   LAST
```

```
    ! M   MIDDLE
    ! N   NONE / NOT A SERIES
    !-->
   <xsd:simpleType name="MessageSequenceEnum">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="F" />
         <xsd:enumeration value="L" />
         <xsd:enumeration value="M" />
         <xsd:enumeration value="N" />
      </xsd:restriction>
   </xsd:simpleType>

   <!--
    ! Enumeration used when "true" and "false" are not expressive
enough, but
    ! use boolean if you can!!!
    !
    ! Y  for true / yes
    ! N  for false / no
    ! U  for unknown
    ! X  for not assigned
    !-->
   <xsd:simpleType name="IndicatorEnum">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="Y" />
         <xsd:enumeration value="N" />
         <xsd:enumeration value="U" />
         <xsd:enumeration value="X" />
      </xsd:restriction>
   </xsd:simpleType>

    <!--
    ! Since R6.
    ! -->
   <xsd:complexType name="Contact">
      <xsd:sequence>
         <xsd:element name="name" type="xsd:string" minOccurs="0" />
         <xsd:element name="phone" type="xsd:string" minOccurs="0" />
         <xsd:element name="email" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
   </xsd:complexType>

</xsd:schema>
```