

Switching for Business Message Specification

Switching for Business (SforB)

Version 3.0

24/10/2025

Contents

1	Introduction	3
1.1	Change log.....	3
1.2	Material changes in Message Specification v0.9.7 relative to v0.0.4	7
1.3	Contributing authors.....	8
1.4	Abbreviations and definitions	8
1.5	Data Types and JSON standards.....	9
1.6	Data Type Specific Requirements.....	9
1.6.1	String	9
1.6.2	URI.....	10
1.6.3	Object.....	10
1.6.4	Array of Objects.....	10
1.6.5	Array of types	10
2	Gaining Provider Lead Business Switching.....	11
2.1	Multi Part Switch Order Processing	11
2.2	Service Identification.....	11
2.2.1	serviceIdentifier	11
3	Business Switching Message Formats.....	12
3.1	Envelope-specific requirements for SforB.....	12
3.2	Business Switch Match Request.....	13
3.2.1	Business Switch Service Match Request	17
3.2.2	Business Switch Match Confirmation.....	18
3.2.3	Business Switch Match Failure	27
3.3	Business Switch Order Request.....	29
3.3.1	Business Switch Order Confirmation.....	30
3.3.2	Business Switch Order Failure	32
3.4	Business Switch Order Update Request	34
3.4.1	Business Switch Order Update Confirmation.....	35
3.4.2	Business Switch Order Update Failure	37
3.5	Business Switch Order Trigger Request.....	39
3.5.1	Business Switch Order Trigger Confirmation.....	40
3.5.2	Business Switch Order Trigger Failure	42
3.6	Business Switch Order Cancellation Request.....	44
3.6.1	Business Switch Order Cancellation Confirmation.....	45
3.6.2	Business Switch Order Cancellation Failure	47
3.7	Message Summary	49

1 Introduction

SforB, Gaining Provider Lead Switching for Business, is a process by which business customers can switch their services from one retailer to another.

This SforB message specification document complements and supports the GPL Business Process Design.

This document provides the messaging specification and any design and development considerations for implementors to support the SforB process.

The core integration elements of the Post Office and Letterbox design, API specifications, security requirements and the messaging envelope are all defined in the One Touch Switch Technical design document. This document focuses only on the One Touch Switch message formats and their use.

For a definition of the SforB process, please refer to the SforB Business Industry Process design document.

1.1 Change log

Version Date Changed by	Reason for change
V0.0.1 First draft 10/03/2023 GPLB-SG	First draft output from the GPLBSG (GPL Switching, Business Steering Group). First draft, specifically providing message formats for the SforB process. Only issued for review within the GPLBSG.
V0.0.2 Updated draft 29/03/2023 GPLB-SG	Added businessSwitchStatusRequest message to the specification for consultation.
V0.0.3 Updated draft for Internal review	Document renamed to GPLB Message Specification Added businessSwitchCarbonCopy message specification for consultation
V0.0.4 Updated draft for Internal review	Fixed multiple inconsistencies and spelling mistakes. Standardised on status names for service actions on confirmed or failed . Removed activation date from businessSwithTriggerRequest as the message itself is the trigger, no pre or post timed messages are permitted.

Version Date Changed by	Reason for change
V0.5 Updated draft 12/10/2023 GPLB-PG	<p>Removed businessSwitchStatusRequest and businessSwitchCarbonCopy. Added businessSwitchNotification. Submitted for public presentation and review.</p> <p>Added businessSwitchServiceMatchRequest as a simple way to add services to an existing match request.</p> <p>Added businessSwitchNotification to provide a mechanism for CPs to contact each other regarding a switch.</p> <p>Renamed activationDate on businessSwitchOrderTriggerRequest to switchDate to remove any inference to behaviour based on the field name.</p> <p>Added name and account as explicit match entries on a businessSwitchMatchRequest as explicit match identifiers.</p> <p>Removed the addressList object from the businessSwitchMatchRequest so only one address object is now present.</p> <p>Renamed linkedSwitchOrderReference to just switchOrderReference on businessSwitchMatchRequest to be consistent with other messages.</p> <p>Defined the assetListRequest/requestedAssets property values on the businessSwitchMatchRequest.</p> <p>Added global allServices as well as individual services to order, update, cancel and switch in the businessSwitchOrderRequest, businessSwitchOrderUpdateRequest, businessSwitchOrderCancelRequest and businessSwitchOrderTriggerRequest.</p>
V0.6 Updated draft 8/11/2023 GPLB-PG	Updated element names to stay aligned with OTS message formats for ease of compatibility
V0.7 Updated for public review 5/12/2023 GPLB-PG	Minor language updates and corrections, submitted for publishing and review.
V0.8 Minor update 18/1/2024 GPLB-PG	Added SecureID as an identification option for switch matches
V0.9 Minor update 1/2/2024 GPLB-PG	<p>Renamed the businessSwitchNotification message to switchNotification so that it can be used and adopted universally for residential as well as business switching (if required).</p> <p>Unified how documents are shared in each message type.</p>
V0.9.1 Minor Update 14/3/2024 GPLB-PG	Added SwitchingNotSupported as a response to a service match request.

Version Date Changed by	Reason for change
V0.9.2 Minor Update 26/3/2024 GPLB-PG	Corrected all example JSON correlation ID values
V0.9.3 Minor Update 23/5/2024 GPLB-SG	Corrected acronyms and tidied presentation (non functional changes)
V0.9.4 Minor Update 1/7/2024 GPLB-SG	Corrected misspellings in businessSwitchOrderUpdateRequest examples.
V0.9.6 Standards 23/9/2024 GPLB-SG	<p>Added section on standards and use of the JSON message.</p> <p>Made document links in messages a generic name to make reuse easier in the schema.</p> <p>Removed no longer supported states for switchAction on the businessSwitchMatchConfirmation message.</p> <p>Made the switch notification message a generic message specification so it can be reused in any messaging protocol, switching or otherwise.</p>
V0.9.7 Vendor neutralised	Remove direct reference to TOTSCo to standardise and equalise the document content.
V0.9.8 Corrections	<p>Corrected example JSON messages using wrong element names.</p> <p>Standardised the Identity names to align with the services, IAS and NBICS instead of just DN.</p> <p>Clarified wording on the use of “port” as an action to confirm it is only for information purposes.</p> <p>Corrected naming of linkedSwitchOrderReference to just switchOrderReference.</p> <p>Removed references to the matching process from the message specification.</p> <p>Removed the matchResult container object as it was no longer needed.</p>
V0.9.9 Proposed changes under review	
V0.9.10 22/5/25 Consolidated changes following review GPLB-PG	<p>Multiple typos, consistency issues corrected. Document cleanup and reformat.</p> <p>Standardises on service confirmation status, confirmed or failed.</p> <p>Removed activationDate from businessSwitchOrderTriggerRequest, the message itself is the trigger and no pre or post dating is allowed.</p>

Version Date Changed by	Reason for change
V0.9.11 21/7/25 Consolidated changes following review GPLB-PG	<p>Made serviceIdentifier mandatory on all messages except match requests, includes responses.</p> <p>Removed linkedMatchReference from businessSwitchMatchRequest.</p> <p>Removed not-sent as a status for asset list and implications of switching sent status.</p> <p>Removed userID and password from all document elements.</p> <p>Increased maximum length of URI elements to 2048 characters.</p> <p>Removed letterOfAuthority from businessSwitchMatchRequest, businessSwitchServiceMatchRequest, and businessSwitchOrderRequest</p> <p>Removed Notification message from the specification.</p>
V0.9.12 1/8/25 Consolidated changes following review GPLB-PG	<p>Removed port as a valid action on all match requests as we have no functional logic based on the value being presented as port and it was causing confusion in the message process.</p> <p>Document renamed and references to GPLB replaced with SforB.</p>
V0.9.13 6/8/25 Final version for release GPLB-PG	<p>Removed action completely from match requests as cease is the only outcome now.</p>
V1.0.0 10/8/25 Published release	
V1.0.1 17/9/25 Correction	<p>Examples of the envelope specification on the businessSwitchMatchConfirmation omitted the required audit data, this has now been added.</p>
V1.02 7/10/25 Correction	<p>On businessSwitchServiceMatchRequest, the switchOrderReference and services properties were listed as optional and should have been required. This has been corrected.</p>
V1.03 15/10/25 Correction	<p>The serviceIdentifiers array in businessSwitchMatchConfirmation has been changed to optional as some products may not have any other useful identifiers, and the actual identity to switch is provided in the separate mandatory serviceIdentifier property. This change aligns with the published Swagger definition.</p> <p>The faultText attributes, for both service/fault and faults/fault has been made mandatory to align with the swagger definition in businessSwitchMatchConfirmation.</p>
V2.0 15/10/25 Up issued for publication	<p>Retitled 'Switching for Business Message Specification' to align with document standard. Headers and footers updated to show correct version, date and 'Public' distribution. <u>No changes to technical information or specification.</u></p>

Version Date Changed by	Reason for change
V3.0 24/10/25 GPLB-PG	Corrected definition table entries for service/identifierType to be service/serviceType on all order, update, trigger and cancel request messages documentation (the examples were correct).

1.2 Material changes in Message Specification v3.0 relative to v0.0.4

[businessSwitchMatchRequest](#), name and account have been added as explicit entities in the root of the message the same as in OTS. [addressList](#) removed so only a single address can be presented on a match request. [letterOfAuthority](#) updated to create a container for the URI and allow supply of a user ID and password for access to avoid documents being available publicly without credentials.

[businessSwitchServiceMatchRequest](#) added to simplify adding services to an existing SOR.

[businessSwitchMatchConfirmation](#), [matchResult](#) added as a container to align with OTS. [SwitchAction](#) expanded to allow [ForcedCease](#) and [ServiceNotUnique](#) as results. [PartialDN](#) added as a value returnable by the LRP if a service is listed that was not requested, for example in the event of identifying a [ForcedCease](#) based on other services switching.

[businessSwitchOrderRequest](#), added [processAllMatchedServices](#) to allow requesting all previously matched services to be ordered without the need to list them individually. Added a default [plannedSwitchDate](#) if services are ordered but not listed explicitly. Moved the [letterOfAuthority](#) document to a dedicated object and included user id and password to control access.

[businessSwitchOrderUpdateRequest](#), added [processAllOrderedServices](#) to allow requesting all previously matched services to be updated without the need to list them individually. Added a default [plannedSwitchDate](#) if services are ordered but not listed explicitly.

[businessSwitchOrderTriggerRequest](#), added [processAllOrderedServices](#) to allow requesting all previously matched services to be triggered without the need to list them individually. Added a default [plannedSwitchDate](#) if services are ordered but not listed explicitly. Added [activationDate](#) to each service as an optional field.

[businessSwitchOrderCancellationRequest](#), added [processAllOrderedServices](#) to allow requesting all previously matched services to be cancelled without the need to list them individually.

Removed the [businessSwitchOrderStatusRequest](#), [businessSwitchOrderOrderStatusReport](#) and [businessSwitchCarbonCopy](#) as no longer required.

Added the [businessSwitchNotification](#) message as a means for CPs to communicate with each other in the event of customer originated issues or disputes.

SecureID has been added as an option for identification in match requests.

Renamed [businessSwitchNotification](#) to [switchNotification](#) to make it a universal process message and not specific to SforB .

Changed the [switchNotification](#) document link to use the same mechanism as the LOA and element structure to standardise handling.

Added the response code [ServiceNotSupported](#) in the Business Switch Match Confirmation message for scenarios where a service was identified correctly by its identifier, but it not supported under the SforB process rules for switching.

All JSON examples have been updated with appropriate and valid correlation ID information.

Removed the [matchResult](#) container as it was not necessary for business switching.

Standardises on service confirmation status, [confirmed](#) or [failed](#).

Removed [activationDate](#) from [businessSwitchOrderTriggerRequest](#), the message itself is the trigger and no pre or post dating is allowed.

Removed [linkedMatchReference](#) from [businessSwitchMatchRequest](#).

Removed [not-sent](#) as a status for asset list and implications of switching sent status. New fault codes will be introduced to communicate the reasons why a communication was not sent.

Removed [userID](#) and [password](#) from all document elements.

Increased maximum length of [URI](#) elements to 2048 characters.

Added the requirement that all messages that specify a [serviceType](#) MUST also specify a unique [serviceIdentifier](#), with the exception of the match request where the value can be left blank in cases of an unknown identifier.

Removed [letterOfAuthority](#) from [businessSwitchMatchRequest](#), [businessSwitchServiceMatchRequest](#), and [businessSwitchOrderRequest](#)

Removed [port](#) from the [action](#) property of switch match messages and amended the explanations as appropriate as the value of [port](#) has no functional purpose in the SforB switching process.

Removed the [action](#) property of switch match messages and amended the explanations as appropriate as the value of [cease](#) was the only valid action remaining.

Made [switchOrderReference](#) and [services](#) required fields on the [businessSwitchServiceMatchRequest](#).

Made the [serviceIdentifiers](#) array in [businessSwitchMatchConfirmation](#) optional.

The [faultText](#) attribute has been made mandatory in all messages where faults are reported.

1.3 Contributing authors

Author	Organisation
Dave Stubbs	Virgin Media

1.4 Abbreviations and definitions

Abbreviation / term	Meaning / definition
MAP	Managed Access Provider. A generic term used to describe any organisation or entity providing connectivity services for the purpose of creating and / or delivering messages for the purposes of SforB, or other processes.
CP	Communications provider This is a term defined by Ofcom in their General Conditions of Entitlement as a “means a person who provides an Electronic Communications Network or an Electronic Communications Service”.
RCP	Retail CP. This term was first defined in the OTS Industry Process (and re-used in the SforB documentation) to define those CPs who provide services at the retail level to end-users, both consumer and business.

Abbreviation / term	Meaning / definition
OTA2	The Office of Telecommunications Adjudicator. The OTA2 was established as a follow-on to the original OTA Scheme, and is independent of Ofcom and of industry.
POTS	Plain old telephone service, a retronym for a voice service over a twisted copper pair – see Plain old telephone service - Wikipedia
SOR	Switch order reference.
SforB	Switching for Business
GPLB-SG	Gaining Provider Led Business Steering Group
GPLB-PG	Gaining Provider Led Business Process Group
SforB-IPD	Switching for Business Industry Process Document

Text in **this font** is a verbatim representation of an element or value that could appear in a SforB message.

1.5 Data Types and JSON standards

The following rules apply to all messages within this specification, but do not cover the rules for the envelope content which are described separately in the JSON Asynchronous Messages specification.

Required values indicate an attribute or structure within the JSON document that must be provided, and must be a minimum of 1 character. An empty string, e.g. "" is not permitted for a required property. A failure to provide a required field may be rejected by a recipient of that message as it would not be in compliance with the message specification. If no content is intended to be passed for a required field, it must at least contain a single space.

Optional values indicate attributes or structures within the JSON document that do not have to be provided unless a value is available or necessary for the purpose of that messages function. Optional values must never be supplied blank unless blank is the intended value of that property. If any validation is applied to an optional field and blank is supplied, then blank must be considered a valid value for that property for it to be accepted. For example, if an account number is specified as optional, if it is provided it must have a value populated and not be left blank. A blank optional field will be treated as a schema violation. If the true intent is to send the message without the account number because it is not available, then the account number element must be omitted from the message.

All structures and arrays if provided must contain at least 1 element, or the structure should be omitted. As an example, if a message is passed and there are no services supplied, then the services structure should be omitted.

1.6 Data Type Specific Requirements

The SforB message specification identifies several data types. Below are the definitions of those datatypes that implementers of the specification should follow to ensure message compliance and process compatibility.

1.6.1 String

A string is a sequence of characters up to 256 characters long unless explicitly stated otherwise.

Some instances of String in the document may specify some precise formatting rules for the strings content, for example a formatted date, others will be free format. Some values may permit or even require longer values, please refer to the schema for the latest definitions.

Where a string of characters can reasonably be more than 256 characters, for example a Free Text value can be as long as 8KB in length, the maximum length will be provided in parenthesis. Message size limitations may apply to long messages, please refer to the industry process and provider API specifications.

Switching for Business Message Specification

1.6.2 URI

A URI is a specially formatted string of characters up to 2048 characters in length conforming to RFC 3986 and providing access to a resource that the recipient of the URI can access.

1.6.3 Object

An object for the purpose of this document is a simple named container and does not have any data content itself.

1.6.4 Array of Objects

An array of objects is a named container in JSON terms that contains a list of objects of the specified type, but those objects are not explicitly named in the JSON message body.

For example, a services array contains a list of service objects, but is encoded as follows.

```
"services": [{
  "serviceType": "IAS",
  "serviceIdentifier": "ABC123",
}, {
  "serviceType": "NBICS",
  "serviceIdentifier": "0101111222",
}, {
  "serviceType": "NBICSRange",
  "serviceIdentifier": "0101112000-0101112999",
}
]
```

In the above example the service objects themselves are not named, but implied based on the services container.

1.6.5 Array of types

Arrays can also be a list of single types, for example a list of URIs can be provided as follows.

```
"documents": [
  "http://mydomain.com/evidence1.pdf",
  "http://mydomain.com/evidence2.pdf"
]
```

2 Gaining Provider Lead Business Switching

Throughout the details of this document, the message formats rely heavily on a clear understanding and communication between the gaining and losing provider of exactly which services are to be switched.

Unlike the consumer One Touch Switch process which is limited to a single broadband instance and a couple of telephone lines at a single site, business switching can mean multiple services across multiple sites on multiple accounts and contracts.

The messaging processes have been designed to allow as much flexibility as necessary to support that level of complexity, while at the same time not being onerous for simpler switches to be processed.

2.1 Multi Part Switch Order Processing

Those familiar with the One Touch Switch process will realise that a single switch has a set pattern of singular events, a match, an order, and a trigger. They must occur in a set order and can only contain the defined set of services and perform the action on those services at each stage.

While that process is lightweight and applicable for consumer switching, it is too rigid to be used in a business environment, and while the high-level concepts of matching, ordering and triggering remain for consistency, the rigidity on the content has been removed to allow for a much more flexible process for business.

This is achieved principally by allowing a switch order to be created, and then allow adding of services to it, then starting the switch on some or all services over a staggered period, and then triggering the switch of those services as and when required. For the period of time that a Switch order is valid, the gaining CP can continue to issue matches, adds, updates, cancellations and triggers associated with that SOR.

When reading this document, you will see that the services affected are listed in every message. This provides the consistency of clear communication required to perform the required actions at each stage of the business switching process.

There are also some helper functions built into the messaging as well, for example on a match request. By allowing a linked match reference to be provided by the GRCP on a switch match request, multiple matches can be combined by the losing provider into a single switch order before a switch order reference is issued.

If there are errors in part on any request, a new request can be submitted only containing the service needing correction. Those already accepted will be actioned.

The design and intent are to allow the business switching process to be flexible, but not unnecessarily complex.

2.2 Service Identification

Service identification is the key to simplification, and there are a number of identities in this document that it is important to clarify and explain before using.

2.2.1 serviceIdentifier

A service identifier is a commonly shared reference for the service that the gaining and the losing provider will both recognise. A telephone number is a good example of this as the UK has a standard format for numbers and they are unambiguous.

A service identifier could identify a single number, or could identify a range, the industry process will specify how these values will be presented.

3 Business Switching Message Formats

The following sections define the suite of messages supported for the SforB process. Each message will be documented with a brief example of the message format, a list of the message elements and their specifications and, if appropriate, their values.

3.1 Envelope-specific requirements for SforB

Every message sent from one CP to another must constitute an envelope and the message body. The purpose of the envelope is to allow MAPs to be able to deliver the message to the correct destination.

For SforB, the elements in the envelope should be populated as follows:

- The source ID should be the RCPID of the sender of the message as it appears in the list obtained from the CPs MAP directory.
- The destination ID should be an RCPID from the list obtained from the CPs MAP directory representing the required recipient.
- The source and destination types should be "RCPID".
- The source correlation ID should be a value generated by the sender of the message, unique to them and for each request made, that would allow any response to be linked to that message. For example UUIDs are recommended.
- The destination correlation ID should only be populated when responding to a message, for example when responding with a success or failure message, and should contain the correlation ID that was provided in the source information of the original message being responded to.
- The **routingID** should be the message format name.
- The **auditData** must be provided in the event of any fault message being returned and must contain the **faultCode**. In the event of a confirmation message containing fault information, only one instance of that message code needs to be included in the audit data, for example if there are 10 mismatched services, all with the same fault code, only one copy of that message should be included in the audit data.

The audit data is provided so that MAPs who are not permitted to process the message body can be informed of any key information necessary for reporting or failure management.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB",
      "correlationID": "d7af2608-108e-4ca2-8162-0b9b796b3a19"
    },
    "routingID": "messageName",
    "auditData": [
      {
        "name": "faultCode",
        "value": "1101"
      }
    ]
  },
  "messageName": {
    ...
  }
}
```

The specification for the above is supplied in the JSON Asynchronous Messaging specification (JAM) v1.0 document available from the OTA website (link to be provided).

3.2 Business Switch Match Request

A business match request defines the information identifying the customer and the services they wish to switch. The following shows the complete message format. For the switching `services` array, at least one service must be specified.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": "businessSwitchMatchRequest"
  },
  "businessSwitchMatchRequest": {
    "grcpBrandName": "Bridgend Telecoms (Telesales)",
    "name": "The Coffee Shop",
    "account": "0003316563216",
    "identity": [{
      "name": "contact",
      "value": "John Smith"
    },
    {
      "name": "NBICS",
      "value": "01234987654"
    },
    {
      "name": "SIC",
      "value": "AB1234"
    }
  ],
  "address": {
    "uprn": "12345",
    "addressLines": ["Unit 1", "Highlands Industrial Estate", "22 Cheshunt Mews"],
    "postTown": "Glasgow",
    "postCode": "SW2R 5AF"
  },
  "switchOrderReference": "ABC46546C46546AAF65AAF65A6E6EA546A4686",
  "assetListRequest": {
    "requester": "John Smith (Accounts)",
    "requestedAssets": "all"
  },
  "services": [{
    "serviceType": "IAS"
  },
  {
    "serviceType": "NBICS",
    "serviceIdentifier": "0101111222"
  },
  {
    "serviceType": "NBICSRange",
    "serviceIdentifier": "0101112000-0101112999"
  }
  ]
}
```

The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchMatchRequest	Identifies the message as being a business switch request.	Object	Required

JSON element	Description	Format	Notes
grcpBrandName	Brand name the gaining provider wishes to be presented to the customer on the implications of switching communication sent to the existing customer.	String	Required
name	This will be either the name of the business switching, or the surname of the customer the account is held under (e.g. as often found on home office accounts).	String	Required
account	The customer account number as known to the losing provider.	String	Optional
Identity	A list of additional identity objects used to help identify the accounts and services being switched	Object of identities	Optional
Identity/name	<p>The name specifies the type of identity information supplied, and therefore indicates how the GRCP should process it.</p> <p>Valid values will be...</p> <p>contact which represents the name of an individual.</p> <p>IAS is an identifier, perhaps shown on the customers bill, that the losing provider would recognise and associate with the account.</p> <p>NBICS which represents a directory number that isn't being switched and is not in the public domain, or immediately associated with the account.</p> <p>SecureID which is a value agreed with the CP to be used when identifying themselves. This could include 2FA codes, etc. subject to security approval.</p> <p>SIC switch identification code. A code the losing CP can use to identify an account. Could be a code on a bill for example. Each CP can choose how they wish to share this code with their customer if they wish to offer that capability.</p>	String	Required
identity/value	This is the value represented by the named identity.	String	Required
address	A container of the address elements	Object of address elements	Required
address/uprn	The UPRN of the premise where the customer is switching services. A UPRN can be up to 12 digits long. For the avoidance of doubt, do not include any leading zeroes.	String	Optional
address/addressLines	Address lines identifying the premise where the switch will take place. Must be formatted according to PAF rules, with no abbreviations. Do not include the post town, postcode or county.	String array	Required
address/postTown	The post town of the address where service is being switched. This should be the post town as defined by	String	Required

JSON element	Description	Format	Notes
	Royal Mail and included in PAF and commercial products based on PAF.		
address/postCode	The post code of the address where service is being switched. Formatted with a space between the inward and outward components.	String	Required
switchOrderReference	Specifies an existing SOR to add this request to. This can be used if more addresses, accounts or services are to be added to an existing switch order and should.	String	Optional
assetListRequest	An instruction to request the losing provider send the requestor a list of the assets that are available for switching in accordance with the identification information presented and the assets required.	Object	Optional
assetListRequest/requester	The identification information to be included with the asset list in cases where the asset list may get sent to a department or location other than where the requester works.	String	Required
assetListRequest/requestedAs sets	Identifies the nature of the assets to be returned, one of the following values are permitted. If this option is not presented then all assets shall be returned. address only list the assets associated with the address presented on this request, this will include other accounts at the same address if the LRCP has that association available. account list the assets associated with the account presented on this request, this will include services at all locations the same account. only list all of the assets associated to the one address and one account provided by the requestor regardless of how many accounts or service locations the LRCP can identify. all list all of the assets associated to all addresses and accounts associated to the requesting customer, as known to the losing provider.	String	Required
services	A container of the services requested to be switched.	Array of services Objects	Optional
service/serviceType	An industry agreed name for the service to be switched. Currently, IAS , NBICS and NBICSRange are supported.	String	Required
service/serviceIdentifier	For services that cannot be identified by just presenting the service type, a serviceIdentifier specifies how the losing provider will find that service this only needs to be used if an asset reference is not provided.	String	Optional

This message is always sent by the gaining provider.

In situations where multiple match requests will be sent for a business switch, for example where there are multiple sites or accounts that are all linked together, the gaining provider will provide either an existing [switchOrderReference](#) on each request. It is always preferable to use an existing [switchOrderReference](#) where one has been obtained instead of creating new match requests. The losing provider will then use this and in each response use the same Switch Order Reference number so that all parts of that switch are combined into a single switching order. For an order placed without a linked match reference, that switch will be processed as a standalone transaction.

All address elements should conform with standard PAF formatting rules in the messages. When processing the request, the losing provider will ignore case sensitivity and must be aware that not all providers will format the address in the same way, some lines may be in different positions, possibly concatenated together, therefore matching is not a simple line-to-line comparison and should allow for variations in formatting.

It is a requirement that abbreviated address elements not be allowed, and that should a losing provider identify abbreviations in the address supplied by the customer they should be expanded to the full words. For example, rd. should always be road, ave. should always be avenue etc. Likewise, gaining providers must have the same consideration if their data stores address elements with abbreviations, or at least accommodate them in their address matching rules.

Each service being switched on behalf of the customer must be listed in the services object, optionally with the action the gaining provider will require to be taken by the losing provider. For the avoidance of doubt, any switched services must be ceased on request regardless of any specified action.

In the event of a successful match, but either no services were provided, or services were provided but not found, this would still be treated as a successful match request, but the failed services would be described in the response body.

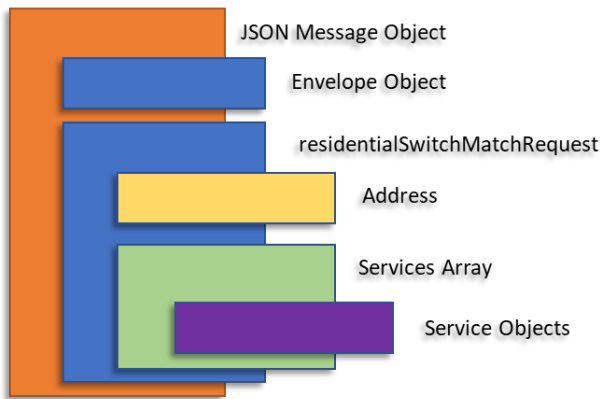
Service identifier is used primarily for voice services to identify the phone number (DN) to be switched, and is not required if the service is the only one present at the specified service location. (rational, in situations where a customer has a phone line but does not know the number...e.g. it was a bundled line they never used, or an old lift line etc).

It is optional against the IAS service in case there are any instances where such an identity would be available and prove useful to match with, otherwise, it can be ignored for that service. The LRCP should use the [serviceIdentifier](#) in an intelligent manner to match against their asset records, e.g:

- For NBICS, it is expected that LRCPs will interpret any [serviceIdentifier](#) as being the DN of the voice service.
- In the event of a number range being required to switch, this can be specified as a single switch service identifier listing the from and to numbers with a hyphen separating them, [01101511000-01101511999](#) for example.

The object structure for this message is as follows.

Switching for Business Message Specification



3.2.1 Business Switch Service Match Request

For an existing switch order reference, to add services, this cut down version of the match request can be used to add services to an existing SOR.

As no further matching is required once an SOR has been issued, this request can be simply used to add services to the existing SOR, for example after requesting an asset list. The replies to this message will still use the standard [businessSwitchMatchConfirmation](#) and [businessSwitchMatchFailure](#) messages.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": "businessSwitchServiceMatchRequest"
  },
  "businessSwitchServiceMatchRequest": {
    "switchOrderReference": "ABC46546C46546AAF65AAF65A6E6EA546A4686",
    "services": [
      {
        "serviceType": "IAS",
        "serviceIdentifier": "ABC123"
      },
      {
        "serviceType": "NBICS",
        "serviceIdentifier": "0101111222"
      },
      {
        "serviceType": "NBICSRange",
        "serviceIdentifier": "0101112000-0101112999"
      },
      {
        "serviceType": "NBICSRange",
        "serviceIdentifier": "0101113000-0101113999"
      }
    ]
  }
}
```

The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchServiceMatch Request	Identifies the message as being a business switch request.	Object	Required
switchOrderReference	Specifies an existing SOR to add this request to.	String	Required

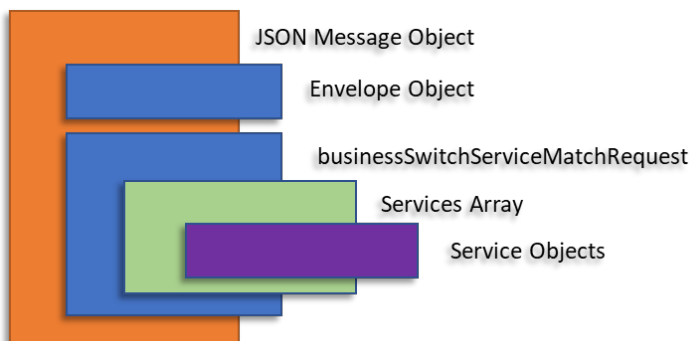
JSON element	Description	Format	Notes
services	A container of the services requested to be switched.	Array of services Objects	Required
service/serviceType	An industry agreed name for the service to be switched. Currently, IAS , NBICS and NBICSRange are supported.	String	Required
service/serviceIdentifier	For services that cannot be identified by just presenting the service type, a serviceIdentifier specifies how the losing provider will find that service this only needs to be used if an asset reference is not provided.	String	Optional

This message is always sent by the gaining provider.

[serviceIdentifier](#) is used primarily for voice services to identify the phone number (DN) to be switched (ported), and is not required if the voice service to be ceased if there is only a single DN on the account being switched. It is also optional against the [IAS](#) service in case there are any instances where such an identity would be available and prove useful to match with, otherwise, it can be ignored for that service. The LRCP should use the [serviceIdentifier](#) in an intelligent manner to match against their asset records, e.g:

- For [NBICS](#), it is expected that most LRCPs will interpret any [serviceIdentifier](#) as being the [DN](#) of the voice service.
- In the event of a number range being required to switch, this can be specified as a single switch service identifier listing the from and to number with a hyphen separating them.
- LRCPs will be required to publish the URL of a help/FAQ page, where they can document what forms of [serviceIdentifier](#) and [type](#) they will accept for IAS (and if they support anything other than [DN](#) on [NBICS](#)).

The object structure for this message is as follows.



3.2.2 Business Switch Match Confirmation

Below is an example of a successful switch match response (not related to the previous matching examples for both the [businessSwitchMatchRequest](#) and the [businessServiceMatchRequest](#)). This message is only ever sent by the losing provider.

```

{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",

```

Switching for Business Message Specification

```
    "identity": "RAAB",
    "correlationID": "ABC123"
  },
  "routingID": "businessSwitchMatchConfirmation",
  "auditData": [
    {
      "name": "faultCode",
      "value": "1191"
    },
    {
      "name": "faultCode",
      "value": "1196"
    },
    {
      "name": "faultCode",
      "value": "1198"
    }
  ]
},
"businessSwitchMatchConfirmation": {
  "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
  "implicationsSent": [{
    "sentMethod": "email",
    "sentTo": "d***@my-domain.com",
    "sentBy": "2022-09-25 09:10:00"
  }, {
    "sentMethod": "2nd class post",
    "sentBy": "2022-09-26 10:00:00"
  }
],
  "assetListSent": [{
    "sentMethod": "email",
    "sentTo": "d***@my-domain.com",
    "sentBy": "2022-09-25 09:10:00"
  }, {
    "sentMethod": "2nd class post",
    "sentBy": "2022-09-26 10:00:00"
  }
],
  "services": [{
    "serviceType": "IAS",
    "switchAction": "ServiceFound",
    "serviceIdentifier": "ABC123",
    "serviceIdentifiers": [{
      "identifierType": "ONTReference",
      "identifier": "123456789"
    }, {
      "identifierType": "PortNumber",
      "identifier": "1"
    }, {
      "identifierType": "NetworkOperator",
      "identifier": "A001"
    }
  ]
}, {
  "serviceType": "NBICS",
  "switchAction": "ServiceFound",
  "serviceIdentifier": "0101111222",
  "serviceIdentifiers": [{
    "identifierType": "CUPID",
    "identifier": "123"
  }, {
    "identifierType": "DN",
    "identifier": "0101111222"
  }
]
}, {
  "serviceType": "NBICSRange",
  "serviceIdentifier": "0101112000-0101112999",
  "switchAction": "ServiceFound",
  "serviceIdentifiers": [{
    "identifierType": "CUPID",
    "identifier": "123"
  }, {
    "identifierType": "DN-From",
    "identifier": "0101112000"
  }
}
```

Switching for Business Message Specification

```

    }, {
      "identifierType": "DN-To",
      "identifier": "0101112999"
    }
  ],
  {
    "serviceType": "NBICSRange",
    "serviceIdentifier": "0101113001-0101113999",
    "switchAction": "ServiceNotFound",
    "fault": {
      "faultCode": "1191",
      "faultText": "Number Range Error"
    }
  }
],
"faults": [{
  "faultCode": "1196",
  "faultText": "Unable to send asset list, account number not found."
},
{
  "faultCode": "1198",
  "faultText": "Unable to determine assets."
}]
}

```

The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchMatchConfirmation	Identifies the message as being a business switch request confirmation, i.e. a positive match response.	Object	Required
switchOrderReference	The switch order reference is a UUID created by the losing provider to represent the match result actions. xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx	String	Required
implicationsSent	A container of the information sent methods..	Array of implications Objects	Optional
implicationsSent/sentMethod	Specifies how information has/will be sent to the customer. Current supported values are email , sms , 1st class post , 2nd class post and verbally . The LRCP must specify at least one method.	String	Required

JSON element	Description	Format	Notes
implicationsSent/sentTo	<p>If the information were sent by email, this property contains an obfuscated copy of the email address. The domain should be fully visible to aid the customer to identify the email hosting provider the email went to. It is recommended to mask all but 2 characters of the email address. If the addressee is 4 or 5 characters, then all but 1 character must be masked, and if the addressee is less than 4 characters then all characters of the addressee must be obfuscated. For example...</p> <p>***@hotmail.com d***@hotmail.com d****@hotmail.com d*****e@hotmail.com d*****y@hotmail.com</p> <p>If the information were sent by SMS, all digits except the last 2 should be obfuscated, for example.</p> <p>*****85 *****67</p>	String	Optional
implicationsSent/sentBy	<p>Specifies the expected date and time the information will be sent to the customer. The time element may be an actual time of dispatch (e.g. for an automatically generated email), or an estimated completion time (e.g. end of batch time for generation of letters), or an end of SLA (e.g. for manually generated email / letter).</p> <p>Formatted as CCYY-MM-DD hh:mm:ss</p>	String	Required
assetListSent	A container of the asset list sent methods..	Array of asset list Objects	Optional
assetListSent/sentMethod	<p>Specifies how the asset list has been sent to the customer. Current supported values are email, sms, 1st class post and 2nd class post. The LRCP must specify at least one method.</p>	String	Required

JSON element	Description	Format	Notes
assetListSent/sentTo	<p>If the asset list were sent by email, this property contains an obfuscated copy of the email address. The domain should be fully visible to aid the customer to identify the email hosting provider the email went to. It is recommended to mask all but 2 characters of the email address. If the addressee is 4 or 5 characters, then all but 1 character must be masked, and if the addressee is less than 4 characters then all characters of the addressee must be obfuscated. For example...</p> <p>***@hotmail.com d***@hotmail.com d****@hotmail.com d*****e@hotmail.com d*****y@hotmail.com</p>	String	Optional
assetListSent/sentBy	<p>Specifies the expected date and time the asset list will be sent to the customer. The time element may be an actual time of dispatch (e.g. for an automatically generated email), or an estimated completion time (e.g. end of batch time for generation of letters), or an end of SLA (e.g. for manually generated email / letter).</p> <p>Formatted as CCYY-MM-DD hh:mm:ss</p>	String	Required
services	A container of the services requested to be switched.	Array of service Objects	Optional
service/serviceType	<p>An industry agreed name for the service to be switched. Currently, IAS, NBICS and NBICSRange are supported.</p> <p>For NBICS, this will be the telephone number in UK format with a leading 0.</p> <p>For NBICSRange, use two NBICS separated with a hyphen, no white spaces.</p>	String	Required
service/serviceIdentifier	<p>This is reflecting back the value passed to provide clear confirmation that a specific service was matched and associated to that reference. If no value was passed, this value must be populated by a value the LRPC considers unique and will expect to receive on all further messages to refer to this service. If the value passed is not considered unique across all services held by the customer, a new value should be returned that is unique.</p>	String	Required

JSON element	Description	Format	Notes
service/switchAction	Either the action the losing provider will take when the switch is performed, or information to the gaining provider about whether the service matched, or not. Please see the definition table for a list of values and their meanings.	String	Required
service/serviceIdentifiers	A list of name and value pairs which the gaining provider can use to identify if an intra-network switch is feasible, and if so, ensure that the correct asset (copper line, full-fibre service) is switched.	Array of service identity objects	Optional
identity/identifierType	<p>For a provided service, the identifier type specifies the nature of that identifier. The type should make sense to a GRCP who use the same networkOperator as the LRCP.</p> <p>For NBICS, DN will be used where the gaining provider had previously supplied the full DN. If the gaining provider had not previously fully identified that service, then PartialDN will be used and a masked copy of the DN will be returned in the identifier. (Openreach's MLPA service exposes an XML tag also called PartialDN for a WLR line belonging to a different CP. All RCPs can return PartialDN where appropriate, even if the voice service is not an Openreach WLR service.)</p> <p>In the case of a switch of a number range, the start and end numbers will be individually listed with the types being "DN-From" and "DN-To".</p> <p>For IAS an identity is not required but must be provided to assist with switching on some networks, e.g. intra Openreach. Please refer to the SforB-IPD for full information.</p> <p>For any PartialDN, the returned number must be masked, only exposing the last 2 digits as follows.. *****43</p>	String	Required
identity/identifier	A value which makes sense in the context of the identifierType .	String	Required
service/fault	In the event of a failure to identify a service, the details will be contained in this object	Object	Optional
fault/faultCode	An error code defining the nature of the error found processing the match request	String	Required
fault/faultText	A human-readable description of the fault code	String	Required
businessSwitchMatch Confirmation/faults	In the event of a successful match, but other actions such as requesting an asset list fail, the details will be contained in this object	Array of fault objects	Optional
faults/fault	An object containing the fault information	Object	Required

JSON element	Description	Format	Notes
fault/faultCode	An error code defining the nature of the error found processing the match request	String	Required
fault/faultText	A human-readable description of the fault code	String	Required

A successful match response contains the details of how and when the implications of switching have been sent to the customer (or the SLA for when they will be sent) and the details of the match concerning the services they have provided.

If a match request was made for multiple services, at least one but not all were found, then the match response will list all the services in the request, identifying which were found and which were not found. The gaining provider can then use this information to either obtain the correct information from the customer or determine if they have the correct provider. The gaining provider can resubmit the request for the incorrect services referencing the Switch Order Reference returned on this response to add the extra services without having to submit all those that had previously matched again.

For each service identified, information about that service will be provided so that the gaining provider can verify the type of switch being performed, especially if they are switching on the same network as this may affect the type of order they may need to raise.

[serviceIdentifiers](#) are optional for IAS, but within specific industry segments, for example for Openreach CPs, this information would be required to assist in creating the appropriate order types targeting the correct asset (copper line or full-fibre service). For voice lines, the [serviceIdentifiers](#) must be returned if the voice service has a physical line (e.g. WLR)¹, but can be omitted on a VoIP service where the customer has not provided the existing DN².

A successful match to a switching request will return a switch order reference that represents the services requested to be switched that were found.

The values that can be returned for the [switchAction](#) attribute are as follows:

Value	Meaning
ServiceFound	The LRCP has found this service for the matched customer. This is a positive result, and the GRCP may raise a switch order asking for this service to be ceased.
ServiceNotFound	The LRCP has not found this service. The overall match may be positive with a SOR (e.g. broadband has been found, but the voice has not).
ServiceAlreadySwitching	This indicates that the service has matched, but that the service is already on a switch order. That order may be from the same gaining CP on a different match order, or from a completely different CP.
ServiceNotUnique	This indicates more than one service matched the information provided and that a unique service identifier is needed to correctly match the specified service.
SwitchingNotSupported	The service specified has been found, but is currently not supported to switch using the SforB processes.

¹ Where the broadband is provided over the same copper line as the WLR, the [serviceIdentifiers](#) returned on IAS and NBICS will be identical.

² If the customer did not provide the DN, we do not want the LRCP to expose the DN (as this could be mis-used as a reverse lookup). It is not necessary to return the DN to proceed with a switch that ceases VoIP service.

The customer must be clearly identified to constitute a valid match confirmation, even if no services are specified or none of the specified services match. This allows switch orders to be built up through use of asset lists or corrective match requests instead of trying to create a single large 100 percent correct match request up front which would be difficult to achieve with complex customers.

If a service has already been requested on the same Switch Order, then [ServiceFound](#) would be the appropriate response.

The LRCP should also return the network operator of the service with another RCP or customer – e.g. in all the examples given, the network operator is [A001](#), which represents Openreach.

The GRCP may ask the customer if they are aware of having services on multiple billing accounts or across multiple RCPs. If the customer is aware, they may be able to provide information for a second match, specifying either a different LRCP, or different customer / account details. If this second match is successful, the GRCP would have enough information to gain the customer’s express consent to all the impacts of switching, and to proceed with multiple switches.

A GRCP may also proceed with a single switch, but must ensure that their systems designers fully understand the combinations, and could defend their decisions to Ofcom. E.g.

- An inter network switch of only IAS could proceed if the NBICS was provided by Openreach as WLR (inferred from the combination of [A001](#) for Openreach and [PartialDN](#)³) – the WLR service with the other RCP or customer could survive the loss of broadband.
- However a cease of a WLR number would mean the forced cessation of any broadband service on the copper line – so this could not proceed without a double match and express customer consent to the double cessation.

If a GRCP performs a match request, and only some of the services matched, perhaps the customer had mistyped a DN, then by using the obtained [switchOrderReference](#) another match request can be submitted for the same site with only the corrected services included and if they match on that request they will be added to the overall switch order. This avoids rejecting the whole order because some items were not found, but adds an amend function effectively to add new services.

Generic Identifier Types

identifierType	Description and example
NetworkOperator	An identifier for the operator of the network on which the service is provided. To be used to assist in helping the gaining provider determine the ordering processes necessary to switch services. See NBICS and IAS tables for variations based on the service type.
UniqueID	An identifier that will identify clearly and uniquely one specific service. This is a required

Identifier Types for NBICS

³ If the NBICS was MPF, the MPF operator would return an Openreach ServiceId for the MPF service. Additionally any broadband on an MPF line has to be with the same MPF CP (and probably on the same customer billing account).

identifierType	Description and example
NetworkOperator	<p>An identifier for the operator of the network on which the service is provided. To be used to assist in helping the gaining provider determine the ordering processes necessary to switch services.</p> <p>RIDs are 3 alpha chars, RCPIDs are 4 alpha chars. Network operators will be identified by A and 3 numbers, e.g. A001. “A” should be memorable as “Access Provider”, and for clarity RCPIDs will not start with A.</p> <p>E.g. if the NBICS is found as WLR, the network operator would be A001 – the value assigned to Openreach.</p> <p>If the NBICS is found as VoIP service, the network operator can be returned as A000, or may be returned as a value representing the VoIP network operator, where it helps the community of RCPs using that network operator to arrange intra-network number transfers.</p>
DN	The UK formatted number is expressed in all digits without spaces. For example 01213339999.
DN-From	In the event a number range has been requested, or a number in a number range, then this value represents the start number in that range that was found.
DN-To	In the event a number range has been requested, or a number in a number range, then this value represents the end number in that range that was found.
PartialDN	The UK formatted number is masked so only the final 2 digits are identifiable. For example 99. PartialDN is an XML tag used by Openreach. All RCPs can return PartialDN where appropriate, even if the voice service is not an Openreach WLR service – e.g. they have found a second voice service, and the partial DN might help the customer to understand which service has been found, without exposing the full DN.
CUPID	<p>Where the response contains a telephone number, the LRCP should return the CUPID of the current voice provider. This can be used by the GRCP in conjunction with their supply chain to determine the lead time for porting (e.g. do all parties support OTS porting).</p> <p>It is expected that all RCPs will either know the CUPID, or will make a real-time call to their supply chain to obtain the CUPID.</p>

Identifier Types for IAS

identifierType	Description and example
NetworkOperator	<p>An identifier for the operator of the network on which the service is provided. To be used to assist in helping the gaining provider determine the ordering processes necessary to switch services.</p> <p>Network operators will be identified by A and 3 numbers, e.g. A001. “A” should be memorable as “Access Provider”.</p>
DN	For NBICS, if a DN had been provided in the matching request, the response may also contain that full DN

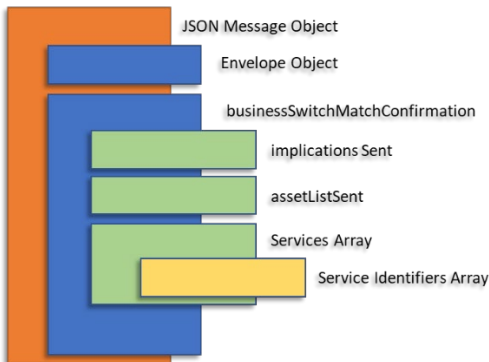
identifierType	Description and example
PartialDN	In scenarios such as broadband over a copper line also used for WLR, the PartialDN of the WLR will also be returned as the serviceIdentifier for the IAS.
ServiceInformation	In the event a customer has more than one IAS instance at the same address under the same account number, then this identifier can be supplied with a plain text value that is human-readable and useful to the customer in clearly identifying the service instance. This could contain the product/speed, the install date, the install location etc. E.g: "10GB FTTP, Installed 25th September 2022 (Comms Room)"

The above values of [identifierType](#) are the generic values that can be used by all RCPs.

The SforB-IPD defines a set of **additional** values to be used by LRCs where the current service(s) are provided on the Openreach network ([AccessLineId](#), [ONTReference](#) and [PortNumber](#)). Other access providers may define their own sets, e.g. where they support intra-network switches.

The [faults](#) section in the [businessSwitchConfirmation](#) message are not for error messages that would reject a match. These messages would be used after a match has been confirmed but a subsequent activity could not be performed, or when secondary data is not valid. For example, if an [assetListRequest](#) has been included, but the information provided is invalid. These conditions should not prevent a match, hence replying with a confirmation, but they do inform the sender they are invalid or could not be carried out as requested.

The object structure for this message is as follows.



3.2.3 Business Switch Match Failure

If the losing provider fails to make a match, the resulting message will represent this through a fault code and description.

```

{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAB",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "routingID": "businessSwitchMatchFailure",
    "auditData": [
      {
        "name": "faultCode",
        "value": "1102"
      }
    ]
  }
}

```

Switching for Business Message Specification

```

    },
    {
      "name": "faultCode",
      "value": "1103"
    }
  ]
},
"businessSwitchMatchFailure": [
  {
    "faultCode": "1102",
    "faultText": "Business name not provided"
  },
  {
    "faultCode": "1103",
    "faultText": "Account not found"
  }
]
}

```

The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchMatchFailure	Identifies the message as being a business switch request failure.	array of fault objects	Required
fault/faultCode	A code defining the nature of the fault found processing the match request	String	Required
fault/faultText	A human-readable description of the fault code	String	Required

This message is returned in response to a switch request where a customer match was not determined.

If any individual element fails to process, unrecognised or invalid values, then the fault should report them as well to assist in diagnostics.

It is important to understand that service mismatches should not result in a failure if the customer was successfully identified.

The object structure for this message is as follows.



3.2.3.1 Response Codes

See the separate “SforB Response Codes” spreadsheet for the list of response codes the losing provider will generate in the event of an error processing a message.

3.3 Business Switch Order Request

At the point a GRCP has approval from a customer to begin switching, the gaining provider will send a business switch order message to signal that the customer has given their express consent to go ahead with this switch, and including the SOR which is to be progressed.

It is not a requirement for the order to contain all services that have previously been matched, only those for which switching is to begin need to be presented, and subsequent order messages can be sent for other services as and when the customer wishes to progress with the switch of those.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": "businessSwitchOrderRequest"
  },
  "businessSwitchOrderRequest": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "plannedSwitchDate": "2025-04-03",
    "processAllMatchedServices": "yes"
  }
}
```

Alternate:

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": "businessSwitchOrderRequest"
  },
  "businessSwitchOrderRequest": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "plannedSwitchDate": "2025-04-03",
    "processAllMatchedServices": "no",
    "services": [{
      "serviceType": "IAS",
      "serviceIdentifier": "ABC123",
    }, {
      "serviceType": "NBICS",
      "serviceIdentifier": "0101111222",
    }, {
      "serviceType": "NBICSRange",
      "serviceIdentifier": "0101112000-0101112999",
      "plannedSwitchDate": "2025-05-27"
    }
  ]
}
```

The services on the switch order request can be omitted and all pending matches will be automatically progressed to order. This makes the process simpler to implement in situations such as small business switches where everything will be switched at once.

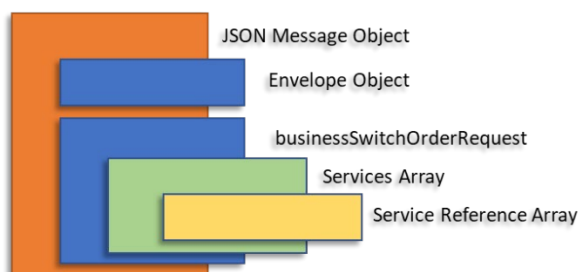
The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchOrderRequest	Identifies the message as being a business switch order.	Object	Required
switchOrderReference	The switch order to be actioned.	String	Required
plannedSwitchDate	The planned switch date represents when the gaining provider expects to deliver service to the customer and is encoded in the format CCYY-MM-DD. At the header level of this message, this refers to any services where no planned switch date have been specified, including any unlisted services requested. It will only be overridden if a specific service is listed with its own plannedSwitchDate.	String	Required
processAllMatchedServices	Valid values are yes or no . If set to yes, any matched service not yet ordered will be considered ordered by the LRCP. Only list individual services in this request if they have a different planned date. If no is specified, then the services must be presented individually in this request.	String	Optional (defaults to no)
services	A container of the services requested to be switched.	Array of services Objects	Optional
service/serviceType	The service type provided by the gaining provider to be switched	String	Required
service/serviceIdentifier	The reference provided by the gaining provider to be switched	String	Required
services/ plannedSwitchDate	The planned switch date represents when the gaining provider expects to deliver service to the customer and is encoded in the format CCYY-MM-DD.	String	Optional

The message presents the switch order reference related to the switching option the customer has given consent to.

The request must either include services to be switched, or the option to **processAllOrderedServices** must be set to **yes**. A combination of these values is also permitted if required.

The object structure for this message is as follows.



3.3.1 Business Switch Order Confirmation

In response to a valid business switch order, the losing provider will reply with a business switch order confirmation.

```
{
```

Switching for Business Message Specification

```

"envelope": {
  "source": {
    "type": "RCPID",
    "identity": "RAAA",
    "correlationID": "XYZ987"
  },
  "destination": {
    "type": "RCPID",
    "identity": "RAAB",
    "correlationID": "ABC123"
  },
  "routingID": "businessSwitchOrderConfirmation",
  "auditData": [
    {
      "name": "faultCode",
      "value": "1217"
    }
  ]
},
"businessSwitchOrderConfirmation": {
  "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
  "services": [
    {
      "serviceType": "IAS",
      "serviceIdentifier": "ABC123",
      "status": "confirmed"
    },
    {
      "serviceType": "NBICS",
      "serviceIdentifier": "0101111222",
      "status": "confirmed"
    },
    {
      "serviceType": "NBICSRange",
      "serviceIdentifier": "0101112000-0101112999",
      "status": "failed",
      "fault": {
        "faultCode": "1217",
        "faultText": "No service match exists"
      }
    }
  ]
}
}

```

The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchOrderConfirmation	Identifies the message as being a business switch order confirmation.	Object	Required
switchOrderReference	The switch order to be actioned.	String	Required
services	A container of the services requested to be switched.	Array of service Objects	Required
service/serviceType	The service type provided by the gaining provider to be switched	String	Required
service/serviceIdentifier	The reference provided by the gaining provider to be switched	String	Required
service/status	The status allows the losing provider to confirm the new state of the specific service in the switching request. Valid values are confirmed meaning the service was ordered successfully, or failed indicating there was problem with the service and the fault block will specify the reason.	String	Required

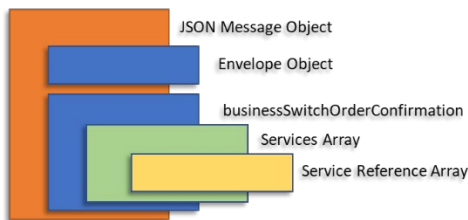
JSON element	Description	Format	Notes
service/fault	In the event of a failure to identify a service, the details will be contained in this object	fault Object	Optional
fault/faultCode	An error code defining the nature of the error found processing the order request	String	Required
fault/faultText	A human-readable description of the fault code	String	Required

The message confirms to the gaining provider that the losing provider has received the order, it is valid and contains validated services that can be ceased when triggered, or if there is an issue with any of the requested service.

The losing provider lists all services affected by the update, including those specifically requested as well as any found if the option to [processAllMatchedServices](#) is set to [yes](#).

Note that the [serviceIdentifier](#) is required for every service, even if one was not originally used on the match request. This is because the value must have been allocated for every service by the LRCP uniquely for the switch order, and avoids any confusion between services. The values will have been returned on the [businessSwitchMatchConfirmation](#).

The object structure for this message is as follows.



3.3.2 Business Switch Order Failure

If the requested switch order reference cannot be found or the services can no longer be switched⁴, then the failure message will be returned.

```

{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAB",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "routingID": "businessSwitchOrderFailure",
    "auditData": [
      {
        "name": "faultCode",
        "value": "1201"
      }
    ]
  },
  "businessSwitchOrderFailure": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "faultCode": "1201",
    "faultText": "Invalid or missing switch order reference"
  }
}

```

⁴ E.g. the services were active when the SOR was generated, but have been ceased since then (e.g. ceased for non payment), and the GRCP attempts a switch order using the SOR.

Switching for Business Message Specification

```
}
```

The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchOrderFailure	Identifies the message as being a business switch order failure.	Object	Required
switchOrderReference	The switch order to be actioned.	String	Required
faultCode	A code defining the nature of the fault found processing the order request	String	Required
faultText	A human-readable description of the fault code	String	Required

The message contains a fault code and description.

If any individual element results in a failure to process or finds unrecognised or invalid values, then the fault should report them as well to assist in diagnostics.

The object structure for this message is as follows.



3.3.2.1 Response Codes

See the separate “SforB Response Codes” spreadsheet for the list of response codes the losing provider will generate in the event of an error processing a message.

3.4 Business Switch Order Update Request

If a gaining provider is taking longer to deliver service and the switch order could expire, then they should notify the losing provider by updating the planned switch date. Please refer to the SforB-IPD for when to send this message.

It is not a requirement for the order to contain all services that have previously been matched, only those for which an updated date need to be provided. This message can be sent as often as required within the defined SLA for SforB.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": "businessSwitchOrderUpdateRequest"
  },
  "businessSwitchOrderUpdateRequest": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "plannedSwitchDate": "2025-04-03",
    "processAllOrderedServices": "yes"
  }
}
```

Alternate:

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": " businessSwitchOrderUpdateRequest "
  },
  "businessSwitchOrderUpdateRequest": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "plannedSwitchDate": "2025-04-03",
    "processAllOrderedServices": "no",
    "services": [{
      "serviceType": "IAS",
      "serviceIdentifier": "ABC123",
    }, {
      "serviceType": "NBICS",
      "serviceIdentifier": "0101111222",
    }, {
      "serviceType": "NBICSRange",
      "serviceIdentifier": "0101112000-0101112999",
      "plannedSwitchDate": "2025-05-27"
    }
  ]
}
```

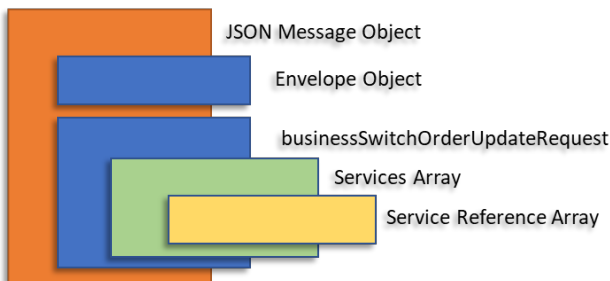
The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchOrderUpdateRequest	Identifies the message as being a business switch order update.	Object	Required

JSON element	Description	Format	Notes
switchOrderReference	The switch order to be actioned.	String	Required
plannedSwitchDate	The planned switch date represents when the gaining provider expects to deliver service to the customer and is encoded in the format CCYY-MM-DD. At the header level of this message, this refers to any services where no planned switch date have been specified, including any unlisted services requested. It will only be overridden if a specific service is listed with its own plannedSwitchDate .	String	Required
processAllOrderedServices	Valid values are yes or no . If set to yes, any ordered service not yet triggered will be considered updated by the LRCP. Only list individual services in this request if they have a different planned date. If no is specified, then the services must be presented individually in this request.	String	Optional (defaults to no)
services	A container of the services requested to be updated.	Array of services Objects	Optional
service/serviceType	The service type provided by the gaining provider to be switched	String	Required
service/serviceIdentifier	The reference provided by the gaining provider to be switched	String	Required
services/ plannedSwitchDate	The planned switch date represents when the gaining provider expects to deliver service to the customer and is encoded in the format CCYY-MM-DD.	String	Optional

The message presents the switch order reference related to the switching option the customer has given consent to. The request must either include services to be updated, or the option to [processAllOrderedServices](#) must be set to [yes](#). A combination of these values is also permitted if required.

The object structure for this message is as follows.



3.4.1 Business Switch Order Update Confirmation

In response to a valid business switch order update, the losing provider will reply with a business switch order update confirmation.

Each service requested to update is individually validated, and even though this is a confirmation request, it may indicate some or all of the services have failed to be identified for updating.

Switching for Business Message Specification

Each service will either contain a response with a status indicating that the service has been confirmed to update, or a fault code and text describing why that service could not be confirmed.

The gaining provider should not take the receipt of the confirmation message as meaning all updates have been confirmed, and should instead inspect every requested service to find out if each has been accepted.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB",
      "correlationID": "ABC123"
    },
    "routingID": "businessSwitchOrderUpdateConfirmation",
    "auditData": [
      {
        "name": "faultCode",
        "value": "1312"
      }
    ]
  },
  "businessSwitchOrderUpdateConfirmation": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "services": [
      {
        "serviceType": "IAS",
        "serviceIdentifier": "ABC123",
        "status": "confirmed"
      },
      {
        "serviceType": "NBICS",
        "serviceIdentifier": "0101111222",
        "status": "confirmed"
      },
      {
        "serviceType": "NBICSRange",
        "serviceIdentifier": "0101112000-0101112999",
        "status": "failed",
        "fault": {
          "faultCode": "1312",
          "faultText": "Service not available to switch"
        }
      }
    ]
  }
}
```

The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchOrderUpdateConfirmation	Identifies the message as being a business switch order update confirmation.	Object	Required
switchOrderReference	The switch order to be actioned.	String	Required
services	A container of the services requested to be updated.	Array of service Objects	Required
service/serviceType	The service type provided by the gaining provider to be switched	String	Required
service/serviceIdentifier	This is the gaining providers switching reference for the service to be switched.	String	Required

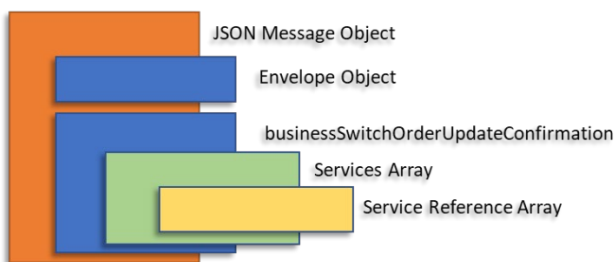
JSON element	Description	Format	Notes
service/status	The status allows the losing provider to confirm the new state of the switching request. The value will be confirmed if successful or failed if the service could not be found or updated and an appropriate fault block will be provided with an explanation of the reason why.	String	Required
service/fault	In the event of a failure to identify a service, the details will be contained in this object	Object	Optional
fault/faultCode	An error code defining the nature of the error found processing the requested service update	String	Required
fault/faultText	A human-readable description of the fault code	String	Required

The message confirms to the gaining provider that the losing provider has received the update order, it is still valid and those validated services can be ceased when triggered, or indicates if there is an issue with the requested service or new date.

Note that the service reference is used, this is because the value has been allocated for every service by the GRCP uniquely for the switch order, and avoids any confusion between asset IDs or service identifiers.

The losing provider lists all services affected by the update, including those specifically requested as well as any found if the option to [processAllOrderedServices](#) is set to [yes](#).

The object structure for this message is as follows.



3.4.2 Business Switch Order Update Failure

If the requested switch order reference cannot be found, or is already fully complete or cancelled, then the failure message will be returned.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAB",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "routingID": "businessSwitchOrderUpdateFailure",
    "auditData": [
      {
        "name": "faultCode",
        "value": "1301"
      }
    ]
  },
  "businessSwitchOrderUpdateFailure": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
  }
}
```

Switching for Business Message Specification

```
    "faultCode": "1301",  
    "faultText": "Invalid or missing switch order reference"  
  }  
}
```

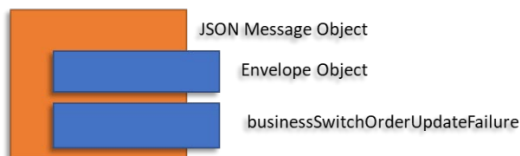
The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchOrderUpdateFailure	Identifies the message as being a business switch order update failure.	Object	Required
switchOrderReference	The switch order to be actioned.	String	Required
faultCode	A code defining the nature of the fault found processing the match request	String	Required
faultText	A human-readable description of the fault code	String	Required

The message contains a fault code and description.

If any individual element results in a failure to process or finds unrecognised or invalid values, then the fault should report them as well to assist in diagnostics.

The object structure for this message is as follows.



3.4.2.1 Response Codes

See the separate “SforB Response Codes” spreadsheet for the list of response codes the losing provider will generate in the event of an error processing a message.

3.5 Business Switch Order Trigger Request

At the point a GRCP has completed the provision of service, they will inform the losing provider by issuing the business switch completion message specifying the services to be ceased.

Again, this message follows the same process as in the switch order and update messages. It is not a requirement for the order trigger to contain all services that have previously been matched, only those for which switching is to be finalised need to be presented, and subsequent trigger messages can be sent for other services as and when the customer wishes to progress with the switch of those.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": "businessSwitchOrderTriggerRequest"
  },
  "businessSwitchOrderTriggerRequest": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "processAllOrderedServices": "yes"
  }
}
```

Alternate:

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": "businessSwitchOrderTriggerRequest"
  },
  "businessSwitchOrderTriggerRequest": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "processAllOrderedServices": "no",
    "services": [
      {
        "serviceType": "IAS",
        "serviceIdentifier": "ABC123"
      },
      {
        "serviceType": "NBICS",
        "serviceIdentifier": "0101111222"
      },
      {
        "serviceType": "NBICSRange",
        "serviceIdentifier": "0101112000-0101112999"
      }
    ]
  }
}
```

The elements of the JSON are defined below:

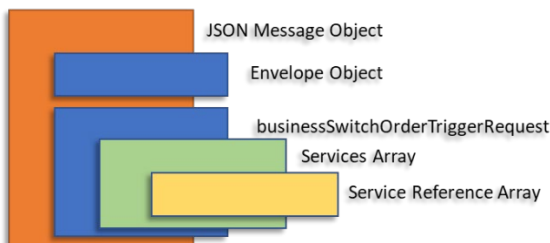
JSON element	Description	Format	Notes
businessSwitchOrderTriggerRequest	Identifies the message as being a business switch order trigger.	Object	Required
switchOrderReference	The switch order reference to be triggered.	String	Required

JSON element	Description	Format	Notes
processAllOrderedServices	Valid values are yes or no . If set to yes, any ordered service not yet triggered will be considered triggered by the LRCP. Only list individual services in this request if they have a different activation date. If no is specified, then the services must be presented individually in this request.	String	Optional (defaults to no)
services	A container of the services requested to be triggered.	Array of services Objects	Optional
service/serviceType	The service type provided by the gaining provider to be switched	String	Required
service/serviceIdentifier	The reference provided by the gaining provider to be switched	String	Required

This message will be sent by the gaining provider for all services to be switched, but can be called multiple times for a switch order if the switch is staged over multiple days/weeks/months.

The date in the trigger message is the last day of rental liability for the LRCP for the services specified (subject to contract conditions). The LRCP may process the message on a later date, but must backdate the last day of rental liability to the date in the trigger message.

The object structure for this message is as follows.



3.5.1 Business Switch Order Trigger Confirmation

In response to a valid business switch order trigger, the losing provider will reply with a business switch order trigger confirmation.

```

{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB",
      "correlationID": "ABC123"
    },
    "routingID": "businessSwitchOrderTriggerConfirmation",
    "auditData": [
      {
        "name": "faultCode",
        "value": "1407"
      }
    ]
  }
}

```

Switching for Business Message Specification

```

    },
    "businessSwitchOrderTriggerConfirmation": {
      "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
      "services": [{
        "serviceType": "IAS",
        "serviceIdentifier": "ABC123",
        "status": "confirmed"
      }, {
        "serviceType": "NBICS",
        "serviceIdentifier": "0101111222",
        "status": "confirmed"
      }, {
        "serviceType": "NBICSRange",
        "serviceIdentifier": "0101112000-0101112999",
        "status": "failed",
        "fault": {
          "faultCode": "1407",
          "faultText": "Service switch not ordered or not recognised"
        }
      }
    ]
  }
}

```

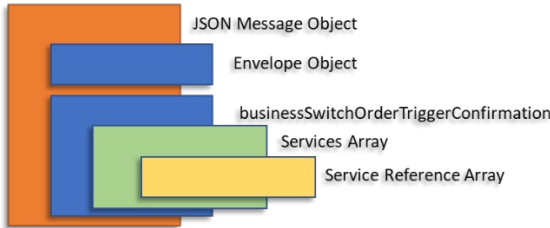
The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchOrderTriggerConfirmation	Identifies the message as being a business switch order trigger confirmation.	Object	Required
switchOrderReference	The switch order to be actioned.	String	Required
services	A container of the services requested to be updated.	Array of service Objects	Required
service/serviceType	The service type provided by the gaining provider to be switched	String	Required
service/serviceIdentifier	This is the gaining providers switching reference for the service to be switched.	String	Required
service/status	The status allows the losing provider to confirm the new state of the switching request. The value will be confirmed if successful or failed If the service could not be found or triggered and an appropriate fault block will be provided with an explanation of the reason why.	String	Required
service/fault	In the event of a failure to identify a service, the details will be contained in this object	Object	Optional
fault/faultCode	An error code defining the nature of the error found processing the requested service update	String	Required
fault/faultText	A human-readable description of the fault code	String	Required

The message confirms to the gaining provider that the losing provider has triggered the order and the actions agreed in the matching request for that order will be carried out.

The object structure for this message is as follows.

Switching for Business Message Specification



3.5.2 Business Switch Order Trigger Failure

If the requested switch order reference cannot be found, or is already complete or cancelled, then the failure message will be returned.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAB",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "routingID": "businessSwitchOrderTriggerFailure",
    "auditData": [
      {
        "name": "faultCode",
        "value": "1401"
      }
    ]
  },
  "businessSwitchOrderTriggerFailure": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "faultCode": "1401",
    "faultText": "Invalid or missing switch order reference"
  }
}
```

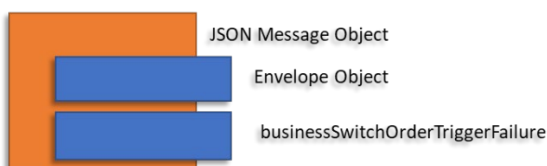
The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchOrderTriggerFailure	Identifies the message as being a business switch order trigger failure.	Object	Required
switchOrderReference	The switch order to be actioned.	String	Required
faultCode	A code defining the nature of the fault found processing the match request	String	Required
faultText	A human-readable description of the fault code	String	Required

The message contains a fault code and description.

If any individual element results in a failure to process or finds unrecognised or invalid values, then the fault should report them as well to assist in diagnostics.

The object structure for this message is as follows.



3.5.2.1 Response Codes

See the separate “SforB Switch Response Codes” spreadsheet for the list of response codes the losing provider will generate in the event of an error processing a message.

3.6 Business Switch Order Cancellation Request

In the event a customer wishes to cancel either an entire switch order, or individual services within a switch order, if there is an active switch in progress, they should notify the losing provider by sending a cancellation request.

It is not a requirement for the order cancellation to contain all services that have previously been ordered to switch, only those for which switching is to be cancelled need to be presented.

In the event all services need to be cancelled, the option to `processAllOrderedServices` set to `yes` can be specified at the request level and omit all unswitched services.

Cancelled services are reset to a matched state, and can be ordered again for switching without the need for a new match request subject to an SLA.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": "businessSwitchOrderCancellationRequest"
  },
  "businessSwitchOrderCancellationRequest": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "processAllOrderedServices": "yes"
  }
}
```

Alternate:

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAB"
    },
    "routingID": "businessSwitchOrderCancellationRequest"
  },
  "businessSwitchOrderCancellationRequest": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "processAllOrderedServices": "no",
    "services": [
      {
        "serviceType": "IAS",
        "serviceIdentifier": "ABC123"
      },
      {
        "serviceType": "NBICS",
        "serviceIdentifier": "0101111222"
      },
      {
        "serviceType": "NBICSRange",
        "serviceIdentifier": "0101112000-0101112999"
      }
    ]
  }
}
```

This message can be sent by the gaining provider any number of times as required for all services currently ordered, or a list of services that are in the ordered state.

Switching for Business Message Specification

The services on the switch order request can be omitted and all pending orders will be cancelled. This makes the process simpler to implement in situations such as small business switches where everything will be switched at once.

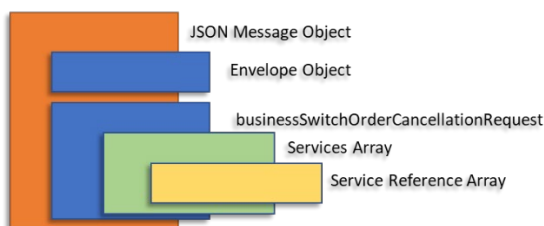
The elements of the JSON are defined below:

JSON element	Description	Format	Notes
businessSwitchOrderCancellationRequest	Identifies the message as being a business switch order cancellation.	Object	Required
switchOrderReference	The switch order reference is to be cancelled.	String	Required
processAllOrderedServices	Valid values are yes or no . If set to yes, any ordered service not yet triggered will be considered cancelled by the LRCP. Only list individual services in this request if all service cancellation is NOT required. If no is specified, then the services must be presented individually in this request.	String	Optional (defaults to no)
services	A container of the services requested to be cancelled.	Array of services Objects	Optional
service/serviceType	The service type provided by the gaining provider to be switched	String	Required
service/serviceIdentifier	The reference provided by the gaining provider to be switched	String	Required

The message presents the switch order reference related to the switch being cancelled. The request must contain at least one service reference to be cancelled.

Any number of service references can be supplied on the request as needed by the GRCP.

The object structure for this message is as follows.



3.6.1 Business Switch Order Cancellation Confirmation

In response to a valid business switch order cancellation, the losing provider will reply with a business switch order cancellation confirmation.

This message has no additional content.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "XYZ987"
    },
    "destination": {
```

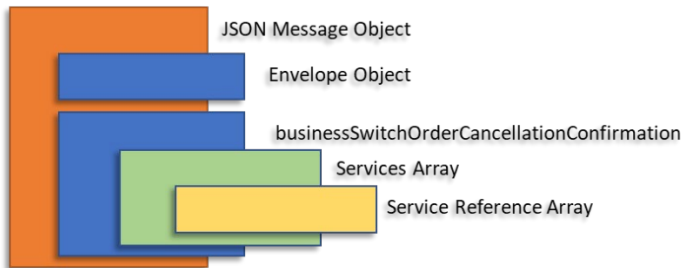

Switching for Business Message Specification

JSON element	Description	Format	Notes
fault/faultText	A human-readable description of the fault code	String	Required

The message confirms to the gaining provider that the losing provider has cancelled the switch order for the specified service references.

Note that the service reference is used, this is because the value has been allocated for every service by the GRCP uniquely for the switch order, and avoids any confusion between asset IDs or service identifiers.

The object structure for this message is as follows.



3.6.2 Business Switch Order Cancellation Failure

If the requested switch order reference cannot be found, or is already complete or cancelled, then the failure message will be returned.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "RAAB",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "RAAA",
      "correlationID": "ABC123"
    },
    "routingID": "businessSwitchOrderCancellationFailure",
    "auditData": [
      {
        "name": "faultCode",
        "value": "1501"
      }
    ]
  },
  "businessSwitchOrderCancellationFailure": {
    "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
    "faultCode": "1501",
    "faultText": "Invalid or missing switch order reference"
  }
}
```

The elements of the JSON are defined below:

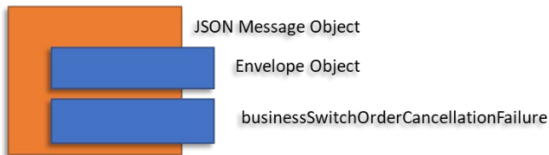
JSON element	Description	Format	Notes
businessSwitchOrderCancellationFailure	Identifies the message as being a business switch request failure.	Object	Required
switchOrderReference	The switch order to be actioned.	String	Required
faultCode	A code defining the nature of the fault found processing the match request	String	Required

JSON element	Description	Format	Notes
faultText	A human-readable description of the fault code	String	Required

The message contains a fault code and description.

If any individual element results in a failure to process or finds unrecognised or invalid values, then the fault should report them as well to assist in diagnostics.

The object structure for this message is as follows.



3.6.2.1 Response Codes

See the separate “Matching Response Codes” spreadsheet for the list of response codes the losing provider will generate in the event of an error processing a message.

3.7 Message Summary

The following table identifies who is responsible for raising each message type:

Gaining Provider	Losing Provider
businessSwitchMatchRequest	businessSwitchMatchConfirmation
businessServiceMatchRequest	businessSwitchMatchFailure
businessSwitchOrderRequest	businessSwitchOrderConfirmation
	businessSwitchOrderFailure
businessSwitchOrderUpdateRequest	businessSwitchOrderUpdateConfirmation
	businessSwitchOrderUpdateFailure
businessSwitchOrderTriggerRequest	businessSwitchOrderTriggerConfirmation
	businessSwitchOrderTriggerFailure
businessSwitchOrderCancellation Request	businessSwitchOrderCancellation Confirmation
	businessSwitchOrderCancellationFailure

End of document