Final draft ETSI ES 202 504-2 V1.1.1 (2008-02)

ETSI Standard

**Open Service Access (OSA);
Parlay X Web Services;
Part 2: Third Party Call
(Parlay X 3)**

Reference

DES/TISPAN-01034-2-OSA

Keywords

API, OSA, service

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Standard (ES) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), and is now submitted for the ETSI standards Membership Approval Procedure.

The present document is part 2 of a multi-part deliverable covering Open Service Access (OSA); Parlay X 3 Web Services, as identified below:

Part 1: "Common";

Part 2: "Third Party Call";

Part 3: "Call Notification";

Part 4: "Short Messaging";

Part 5: "Multimedia Messaging";

Part 6: "Payment";

Part 7: "Account Management";

Part 8: "Terminal Status";

Part 9: "Terminal Location";

Part 10: "Call Handling";

Part 11: "Audio Call";

Part 12: "Multimedia Conference";

Part 13: "Address List Management";

Part 14: "Presence";

Part 15: "Message Broadcast";

Part 16: "Geocoding";

Part 17: "Application-driven Quality of Service (QoS)";

Part 18: "Device Capabilities and Configuration";

Part 19: "Multimedia Streaming Control";

Part 20: "Multimedia Multicast Session Management".

The present document is equivalent to 3GPP TS 29.199-02 V7.4.0 (Release 7).

1 Scope

The present document is part 2 of the Stage 3 Parlay X 3 Web Services specification for Open Service Access (OSA).

The OSA specifications define an architecture that enables application developers to make use of network functionality through an open standardized interface, i.e. the OSA APIs.

The present document specifies the Third Party Call Web Service. The following are defined here:

- Name spaces.
- Sequence diagrams.
- Data definitions.
- Interface specification plus detailed method descriptions.
- Fault definitions.
- Service Policies.
- WSDL Description of the interfaces.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
    - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
    - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] W3C Recommendation (2 May 2001): "XML Schema Part 2: Datatypes".

NOTE: Available at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

[2] ETSI ES 202 504-1: "TISPAN; Open Service Access (OSA); Parlay X Web Services; Part 1: Common (Parlay X 3)".

[3] ETSI ES 202 504-12: "TISPAN; Open Service Access (OSA); Parlay X Web Services; Part 12: Multimedia Conference (Parlay X 3)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 202 504-1 [2] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 202 504-1 [2] apply.

4 Detailed service description

Currently, in order to perform a third party call in telecommunication networks we have to write applications using specific protocols to access Call Control functions provided by network elements (specifically operations to initiate a call from applications). This approach requires a high degree of network expertise. We can also use the OSA gateway approach, invoking standard interfaces to gain access to call control capabilities, but these interfaces are usually perceived to be quite complex by application IT developers. Developers must have advanced telecommunication skills to use Call Control OSA interfaces.

In this clause we describe a Parlay X 3 Web Service, Third Party Call, for creating and managing a call initiated by an application (third party call). The overall scope of this Web Service is to provide functions to application developers to create a call in a simple way. Using the Third Party Call Web Service, application developers can invoke call handling functions without detailed telecommunication knowledge.

The underlying model of the service is based on the following entities:

- **Call Session:** a call (uniquely identified) to which participants can be added/removed.
- **Call Participant:** each of the call parties (uniquely identified) involved in the call session.
- **Media**: the call can utilize multiple media types to support the participants' communication. In particular both audio and video streams are available, including the specific stream direction (i.e. incoming, outgoing, bidirectional).

NOTE 1: Call participants in a Call Session are anticipated to be uniquely identifiable using their URI address.

An application setting up a call session must initially invoke the **makeCallSession**. The result of such invocation is the creation of a "context" that represents a call session with usually two participants, or at a minimum one participant connected; a unique identifier is assigned to the just-created call session. Subsequently the application may wish to add, remove, park or transfer call participants. In order to do so the operations **addCallParticipant, transferCallParticipant, deleteCallParticipant** can be used. Furthermore the call session or call participant status including the media details can be read. In order to do so the operations **getCallParticipantInformation,** and **getCallSessionInformation** can be used. It is also possible to retrieve only the media details using the **getMediaForParticipant** or **getMediaForCall** operations of the Audio Call web service.

The application can also force the call session and all its participants to be terminated with the operation **endCallSession**.

NOTE 2: A call session allows the application to avail of other web service features that can add value to the created call session. For example the Audio Call web service can provide multimedia message delivery to call participants in the call session (**playXxxMessage** operations) and furthermore control of the media types for the call participants thus enabling conversational multimedia communication including voice, video, chat, and data. Media can be added/removed for each participant.

Figure 1 shows a scenario using the Third Party Call Web Service to handle third party call functions. The application invokes a Web Service to retrieve stock quotes and a Parlay X Interface to initiate a third party call between a broker and his client.

In the scenario, whenever a particular stock quote reaches a threshold value (1) and (2), the client application invokes a third party call between one or more brokers and their corresponding customers to decide actions to be taken. After invocation (3) by the application, the Third Party Call Web Service invokes a Parlay API method (4) using the Parlay/OSA SCS-CC (Call control) interface. This SCS handles the invocation and sends a message (5) to an MSC to set-up a call between user A and user B.

In an alternative scenario, the Parlay API interaction involving steps (4) and (5) could be replaced with a direct interaction between the Third Party Call Web Service and the Mobile network.

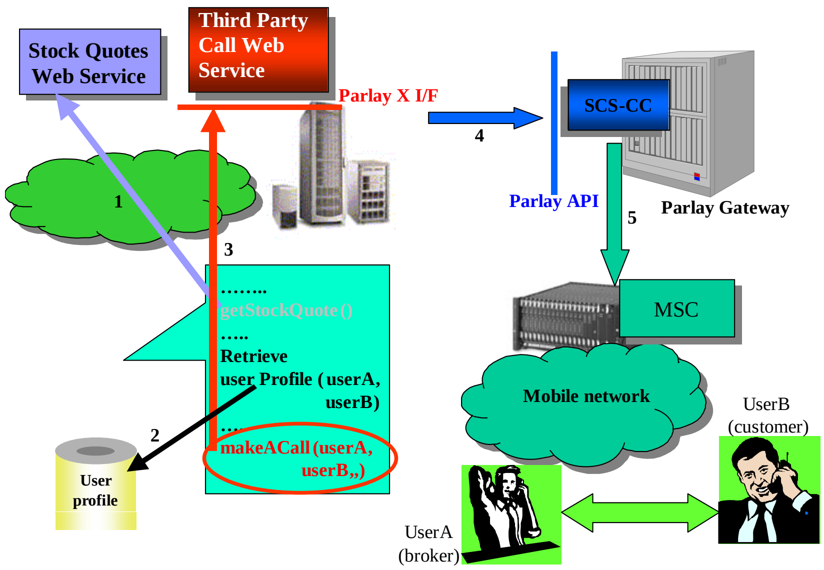

Figure 1: Third party call scenario

5 Namespaces

The ThirdPartyCall interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/third_party_call/v3_4

The 'xsd' namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [1]. The use of the name 'xsd' is not semantically significant.

6 Sequence diagrams

6.1 'Click to Dial' call setup

A common convergence application is Click to Dial, where a self service portal provides a web page that can initiate a call between two phones. This sequence shows a basic call setup, and ending the call through the portal.

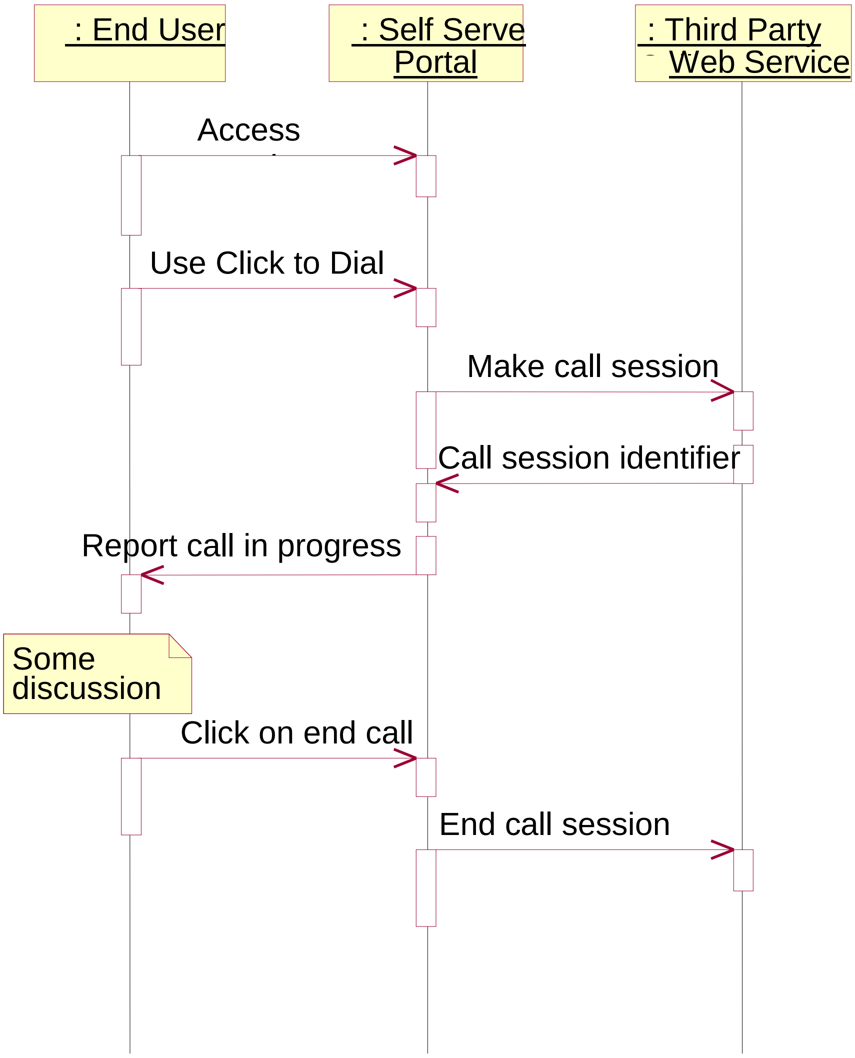

Figure 2

7 XML Schema data type definition

None.

8 Web Service interface definition

8.1 Interface: ThirdPartyCall

This interface provides the ability to setup a call session, add and delete a call participant, transfer a call participant from one call session into another call session, determine the status of an individual call participant or a complete call session, and finally to end a call session.

8.1.1 Operation: makeCallSession

The invocation of **makeCallSession** requests to set-up a call session between two addresses, a first **callParticipant** ("A-Party") and a second **callParticipant** ("B-Party"), provided that the invoking application is allowed to connect them. Optionally, the caller's ("A-Party's") name is provided. Optionally a call session with only a single **callParticipant** ("A-party") can be set-up, for example to play media to the call participant using Audio Call web service capabilities (e.g. **PlayMedia** interface). Optionally the application can also indicate the charging information (**charging**). Optionally, the media type(s) to be used for the participants in the call session can be requested (**mediaInfo**). A service policy details if multimedia application control is supported. If the **mediaInfo** part is omitted, the media type(s) shall be negotiated by the underlying network. A **callSessionIdentifier** is returned on invocation of the **makeCallSession** operation. This **callSessionIdentifier** may be used to retrieve the call session status for all the call participants, including their media type information, using the **getCallSessionInformation** operation. Alternatively, this information may be retrieved for a single call participant using the **getCallParticipantInformation** operation.

During call set-up, the first **callParticipant** ("A-Party") might wait for the second **callParticipant** ("B-Party") to answer the call hearing ringback tone. The **makeCallSession** operation creates a call session with one or two call participants and generates a new **callSessionIdentifier**, which is identified by the string returned in the **makeCallSessionResponse** message. The operation completes as soon as the request is received by the system, i.e. the actual call session is established asynchronously.

In order to receive the information on call status the application has to explicitly invoke the **getCallSessionInformation** operation using the **callSessionIdentifier** returned.

8.1.1.1 Input message: makeCallSessionRequest

Part name	Part type	Optional	Description
callParticipants	xsd:anyURI [1..2]	No	It contains the address of the first participant, and if supplied the second participant, involved in the call
callingParticipantName	xsd:string	Yes	It contains the name of the caller, e.g. the name on whose behalf the call session is being established.
charging	common:Charging Information	Yes	Charge to apply to the call session
mediaInfo	common:MediaInfo [0..unbounded]	Yes	It identifies one or more media type(s) for the call, i.e. the media type(s) to be applied to the participants in the call session. For each media type the media direction - incoming, outgoing, or bidirectional - shall be indicated. An empty array shall have the same meaning as if the part is omitted. If this part is omitted, the media type(s) shall be negotiated by the underlying network.
changeMediaNot Allowed	xsd:boolean	No	If true, no call participant (user) in the call will be permitted to change media type during the call. If false the end user may change media type after the call is established as no network protection mechanism is set up to prevent participant (end user) initiated change of media type.

8.1.1.2 Output message: makeCallSessionResponse

Part name	Part type	Optional	Description
result	xsd:string	No	It identifies the specific call session created

8.1.1.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error.

- POL0008 - Charging not supported.
- POL0011 – Media type not supported

8.1.1a Operation: addCallParticipant

The invocation of **addCallParticipant** requests to add a call participant to an existing call session.
The **addCallParticipantRequest** operation has two mandatory parts, the first is the identifier of the call session to which the participants should be added, and the second is the participant to add (contains the URI of the participant).

Optionally, the media type(s) that shall be added for the specific call participant may be requested using the **mediaInfo** part. If the part is omitted, the media type(s) shall be negotiated by the underlying network. The call session or call participant status, including information on media types, can be retrieved using the **getCallParticipantInformation** and **getCallSessionInformation** operations.

All occurrences of invalid call session or participant address shall result in an invalid input value exception.

8.1.1a.1 Input message: addCallParticipantRequest

Part name	Part type	Optional	Description
callSessionIdentifier	xsd:string	No	It identifies the existing call session. This must be a non-null value as it identifies a pre-existing call session in the network
callParticipant	xsd:anyURI	No	It contains the address of the user to add to the existing call session identified by the **callSessionIdentifier**
mediaInfo	common:MediaInfo [0..unbounded]	Yes	It identifies one or more media type(s) for the participant to be added to the call session. For each media type the media direction - incoming, outgoing, or bidirectional - shall be indicated. An empty array shall have the same meaning as if the part is omitted. If this part is omitted, the media type(s) shall be negotiated by the underlying network.

8.1.1a.2 Output message: addCallParticipantResponse

Part name	Part type	Optional	Description
None			

8.1.1a.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0261 - Call session already terminated.

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error.
- POL0011 - Media type not supported.
- POL0240 - Too many participants. - from ES 202 504-12 [4].

8.1.1b Operation: transferCallParticipant

The invocation of **transferCallParticipant** enables a call transfer, effectively transferring a call participant from one call session into another call session.

The **transferCallParticipantRequest** message has three parts, the first is the call session identifier (destination call session) to where the participants should be moved, the second is the source call session where the participant to transfer is currently located, and the third identifies the call participant to transfer (contains the URL of the participant). Upon completion of the **transferCallParticipant** operation, the call participant is implicitly deleted from the source call session.

In transferring the call participant to the new destination call session, the participant information is not affected. When, as a result of transferring a call participant, the source call session is left without any call participant, the source call session is terminated.

All occurrences of invalid call session or participant address shall result in an invalid input value exception.

8.1.1b.1 Input message: transferCallParticipantRequest

Part name	Part type	Optional	Description
destinationCallSessionIdentifier	xsd:string	No	It identifies the existing target call session to which the call participant to transfer shall be moved.
sourceCallSessionIdentifier	xsd:string	No	It identifies the existing source call session, which contains the call participant to transfer.
callParticipant	xsd:anyURI	No	This is the address of the participant to be transferred from the source call session, which is identified by the **sourceCallSessionIdentifier**.

8.1.1b.2 Output message: transferCallParticipantResponse

Part name	Part type	Optional	Description
None			

8.1.1b.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0261 - Call session already terminated.

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error.
- POL0240 - Too many participants. - from ES 202 504-12 [3].

8.1.2 Operation: getCallParticipantInformation

The invocation of **getCallParticipantInformation** retrieves the current call participant status, of type **CallParticipantInformation**, for the call participant identified by **callParticipant,** within the call session identified by the **callSessionIdentifier**. This operation can be invoked multiple times by the application even if the call has already ended. However, after the call has ended, status information will be available only for a limited period of time that is specified in the service policy 'StatusRetentionTime'.

The **getCallParticipantInformationRequest** message has two parts, the first is the call session identifier where the participant's information should be retrieved and the second is the call participant identifier (contains the URL of the participant).

All occurrences of invalid call session or call participant address shall result in an invalid input value exception.

8.1.2.1 Input message: getCallParticipantInformationRequest

Part name	Part type	Optional	Description
callSessionIdentifier	xsd:string	No	It identifies the specific call session
callParticipant	xsd:anyURI	No	It identifies a specific call participant address within the call session

8.1.2.2 Output message: getCallParticipantInformationResponse

Part name	Part type	Optional	Description
result	common:CallParticipant Information	No	It identifies the status of the requested call participant.

8.1.2.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0261 - Call session already terminated.

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error.

8.1.3 Void

8.1.4 Void

8.1.5 Operation: getCallSessionInformation

The invocation of **getCallSessionInformation** retrieves information associated with the call session identified by the c**allSessionIdentifier** message part. The information retrieved includes the call participant information. This operation can be invoked multiple times by the application even if the session has already ended. However, after the session has ended, status information will be available only for a limited period of time that is specified in the service policy 'StatusRetentionTime'.

All occurrences of invalid call session shall result in an invalid input value exception

8.1.5.1 Input message: getCallSessionInformationRequest

Part name	Part type	Optional	Description
callSessionIdentifier	xsd:string	No	It identifies a specific call session

8.1.5.2 Output message: getCallSessionInformationResponse

Part name	Part type	Optional	Description
result	common:CallSession	No	It identifies the status of the session

8.1.5.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error.

- SVC0002 - Invalid input value.
- SVC0261 - Call session already terminated.

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error.

8.1.6 Operation: deleteCallParticipant

The invocation of **deleteCallParticipant** removes the call participant identified by **callParticipant** from the call session identified by **callSessionIdentifier,** and implicitly terminates that participant's involvement in the call session.

The **deleteCallParticipantRequest** message has two parts, the first is the call session identifier where the participant to delete is located and the second part identifies the call participant to be deleted (contains the URL of the participant).

All occurrences of invalid call session or participant address shall result in an invalid input value exception.

8.1.6.1 Input message: deleteCallParticipantRequest

Part name	Part type	Optional	Description
callSessionIdentifier	xsd:string	No	It identifies a specific call session
callParticipant	xsd:anyURI	No	It identifies a specific call participant within the identified call session.

8.1.6.2 Output message: deleteCallParticipantResponse

Part name	Part type	Optional	Description
None			

8.1.6.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0261 - Call session already terminated.

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error.

8.1.7 Operation: endCallSession

The invocation of **endCallSession** terminates the call session identified by **callSessionIdentifier**.

The call to all participants is ended.

All occurrences of invalid call session shall result in an invalid input value exception.

8.1.7.1 Input message: endCallSessionRequest

Part name	Part type	Optional	Description
callSessionIdentifier	xsd:string	No	It identifies a specific call session

8.1.7.2 Output message: endCallSessionResponse

Part name	Part type	Optional	Description
None			

8.1.7.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0261 - Call session already terminated.

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error.

9 Fault definitions

The following faults are defined for this service.

9.1 ServiceException

9.1.1 SVC0260: Void

9.1.2 SVC0261: Call session already terminated

Part name	Description
messageId	SVC0261
text	Call session has already been terminated
variables	None

10 Service policies

These service policies are defined for the Third Party Call service.

Name	Type	Description
ChargingAllowed	xsd:boolean	Indicates whether charging is allowed for the **makeCallSession** operation
StatusRetentionTime	common: TimeMetric	Length of time to retain status after the termination of the call
ChangeMediaNotAllowed	xsd:boolean	Indicates whether an end user can change the media types used in a call.
MaximumParticipants	xsd:int	Maximum number of participants which a session can serve.
MultimediaSupported	xsd:boolean	Indicates whether multimedia is supported and whether an application can change the media types used in a call.

Annex A (normative): WSDL for Third Party Call

The document/literal WSDL representation of this interface specification is compliant to ES 202 504-1 [2] and is contained in text files (contained in archive es_20250402v010101m0.zip) which accompany the present document.

Annex B (informative): Bibliography

ETSI TR 121 905: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

History

Document history		
V1.1.1	February 2008	Membership Approval Procedure MV 20080425: 2008-02-26 to 2008-04-25