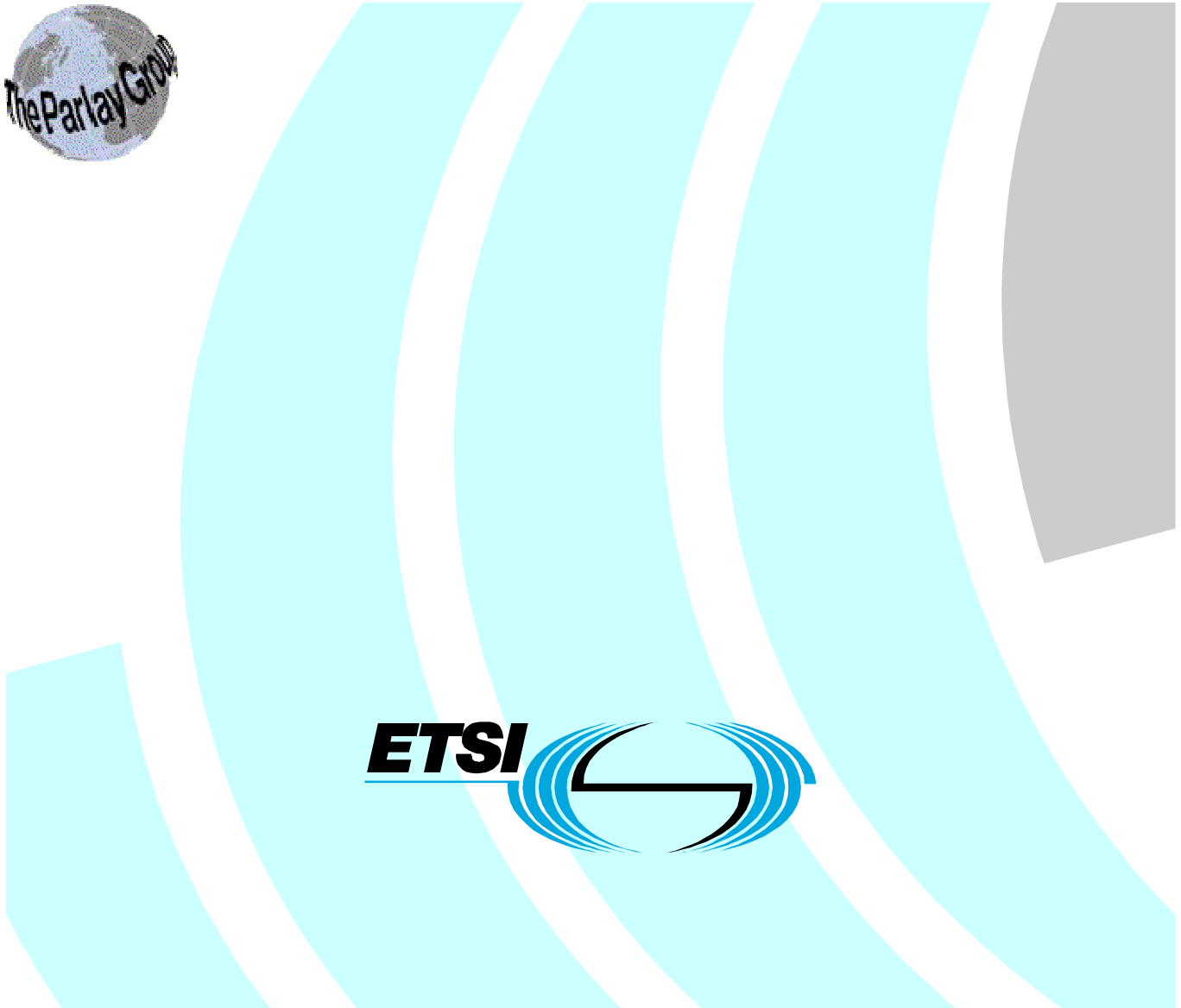
ETSI TR 102 397-7 V1.1.1 (2005-08)

Technical Report

Open Service Access (OSA);
Mapping of Parlay X Web Services to Parlay/OSA APIs;
Part 7: Account Management Mapping



Reference
DTR/TISPAN-01021-07-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 5
Foreword 5
1 Scope 6
2 References 6
3 Definitions and abbreviations 6
3.1 Definitions 6
3.2 Abbreviations 6
4 Mapping description 6
5 Sequence diagrams 7
5.1 Query account balance 7
5.2 Retrieve account transaction history 7
5.3 Query account credit expiry date (Parlay 5.0 only) 8
5.4 Increment account balance (Parlay 5.0 only) 8
5.5 Increment account balance using Voucher (Parlay 5.0 only) 9
6 Detailed mapping information 9
6.1 Operations 9
6.1.1 getBalance 9
6.1.1.1 Mapping to `IpAccountManager.queryBalanceReq` 9
6.1.1.2 Mapping from `IpAppAccountManager.queryBalanceRes` 10
6.1.1.3 Mapping from `IpAppAccountManager.queryBalanceErr` 10
6.1.2 getHistory 10
6.1.2.1 Mapping to `IpAccountManager.retrieveTransactionHistoryReq` 11
6.1.2.2 Mapping from `IpAppAccountManager.retrieveTransactionHistoryRes` 11
6.1.2.3 Mapping from `IpAppAccountManager.retrieveTransactionHistoryErr` 11
6.1.3 getCreditExpiryDate (Parlay 5.0 only) 11
6.1.3.1 Mapping to `IpAccountManager.queryBalanceExpiryDateReq` 12
6.1.3.2 Mapping from `IpAppAccountManager.queryBalanceExpiryDateRes` 12
6.1.3.3 Mapping from `IpAppAccountManager.queryBalanceExpiryDateErr` 12
6.1.4 balanceUpdate (Parlay 5.0 only) 12
6.1.4.1 Mapping to `IpAccountManager.updateBalanceReq` 13
6.1.4.2 Mapping from `IpAppAccountManager.updateBalanceRes` 13
6.1.4.3 Mapping from `IpAppAccountManager.updateBalanceErr` 13
6.1.5 voucherUpdate (Parlay 5.0 only) 13
6.1.5.1 Mapping to `IpAccountManager.queryUserVouchersReq` 14
6.1.5.2 Mapping from `IpAppAccountManager.queryUserVouchersRes` 14
6.1.5.3 Mapping from `IpAppAccountManager.queryUserVouchersErr` 14
6.1.5.4 Mapping to `IpAccountManager.queryVoucherReq` 14
6.1.5.5 Mapping from `IpAppAccountManager.queryVoucherRes` 15
6.1.5.6 Mapping from `IpAppAccountManager.queryVoucherErr` 15
6.1.5.7 Mapping to `IpAccountManager.updateBalanceReq` 15
6.1.5.8 Mapping from `IpAppAccountManager.updateBalanceRes` 15
6.1.5.9 Mapping from `IpAppAccountManager.updateBalanceErr` 15
6.1.5.10 Mapping to `IpAccountManager.destroyVoucherReq` 16
6.1.5.11 Unmapped Elements 16
6.2 Exceptions 16
6.2.1 Mapping from `TpBalanceQueryError` 16
6.2.2 Mapping from `TpTransactionHistoryStatus` 16
6.2.3 Mapping from `TpVoucherError` 16
6.2.4 Mapping from Parlay/OSA method exceptions 17

7 Additional notes 17
History 18

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 Technical Report (TR) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN).

The present document is part 7 of a multi-part deliverable covering Open Service Access (OSA); Mapping of Parlay X Web Services to Parlay/OSA APIs, as identified below:

Part 1: "Common Mapping";

Part 2: "Third Party Call Mapping";

Part 3: "Call Notification Mapping";

Part 4: "Short Messaging Mapping";

Part 5: "Multimedia Messaging Mapping";

Part 6: "Payment Mapping";

Part 7: "Account Management Mapping";

Part 8: "Terminal Status Mapping";

Part 9: "Terminal Location Mapping";

Part 10: "Call Handling Mapping";

Part 11: "Audio Call Mapping";

Part 12: "Multimedia Conference Mapping";

Part 14: "Presence Mapping";

NOTE: Part 13 has not been provided as there is currently no defined mapping between ES 202 391-13 [4] and the Parlay/OSA APIs. If a mapping is developed, it will become part 13 of this series.

The present document has been defined jointly between ETSI, The Parlay Group (http://www.parlay.org) and the 3GPP.

1 Scope

The present document specifies the mapping of the Parlay X Account Management Web Service to the Account Management Service Capability Feature (SCF).

The Parlay X Web Services provide powerful yet simple, highly abstracted, imaginative, telecommunications functions that application developers and the IT community can both quickly comprehend and use to generate new, innovative applications.

The Open Service Access (OSA) specifications define an architecture that enables application developers to make use of network functionality through an open standardized interface, i.e. the Parlay/OSA APIs.

2 References

For the purposes of this Technical Report (TR), the following references apply:

[1] ETSI TR 121 905: "Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

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

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

[3] ETSI TR 102 397-1: " Open Service Access (OSA); Mapping of Parlay X Web Services to Parlay/OSA APIs; Part 1: Common Mapping".

[4] ETSI ES 202 391-13: "Open Service Access (OSA); Parlay X Web Services; Part 13: Address List Management".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 102 397-1 [3] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 102 397-1 [3] apply.

4 Mapping description

The Account Management capability can be implemented with Parlay/OSA Account Management.

It is applicable to ETSI OSA 1.x/2.x/3.x, Parlay/OSA 3.x/4.x/5.x and 3GPP Releases 4 to 6.

Some Account Management web service operations cannot be mapped to Parlay/OSA releases earlier than ETSI 3.x/ Parlay 5.x/ 3GPP 6.x . Where applicable, this constraint is identified in clauses 5 and 6.

5 Sequence diagrams

5.1 Query account balance

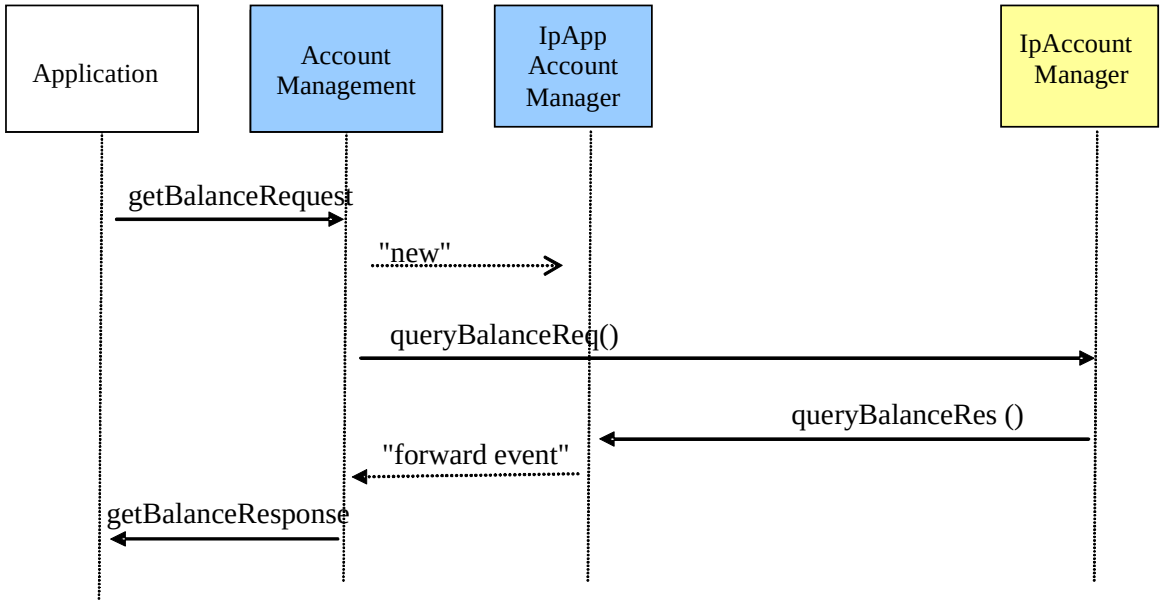


Figure 1

5.2 Retrieve account transaction history

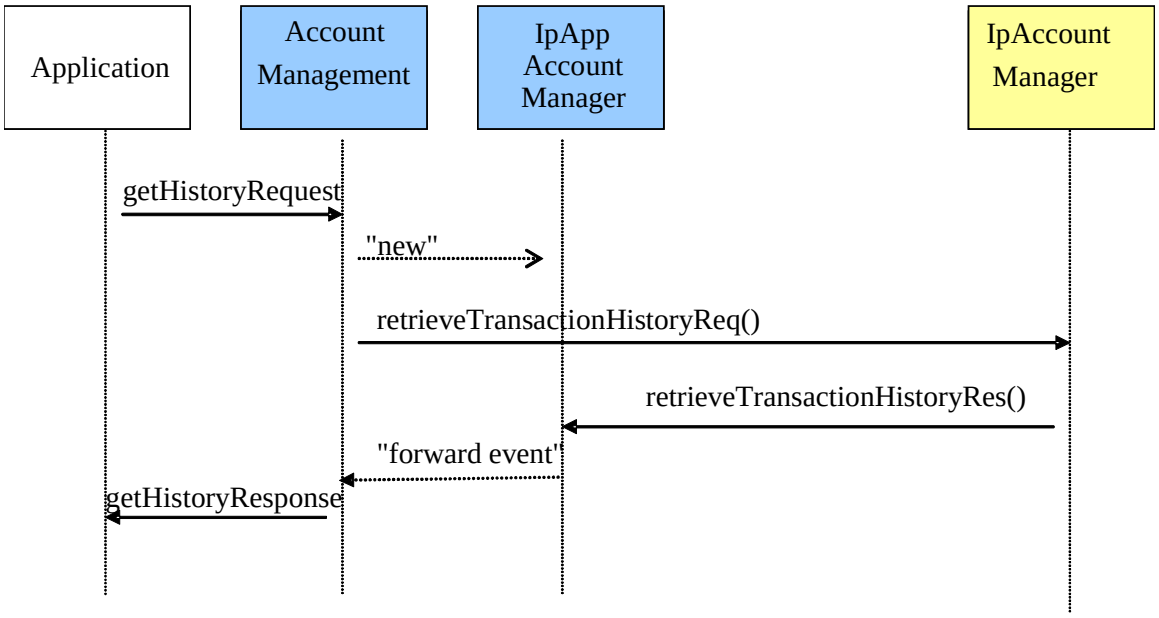


Figure 2

5.3 Query account credit expiry date (Parlay 5.0 only)

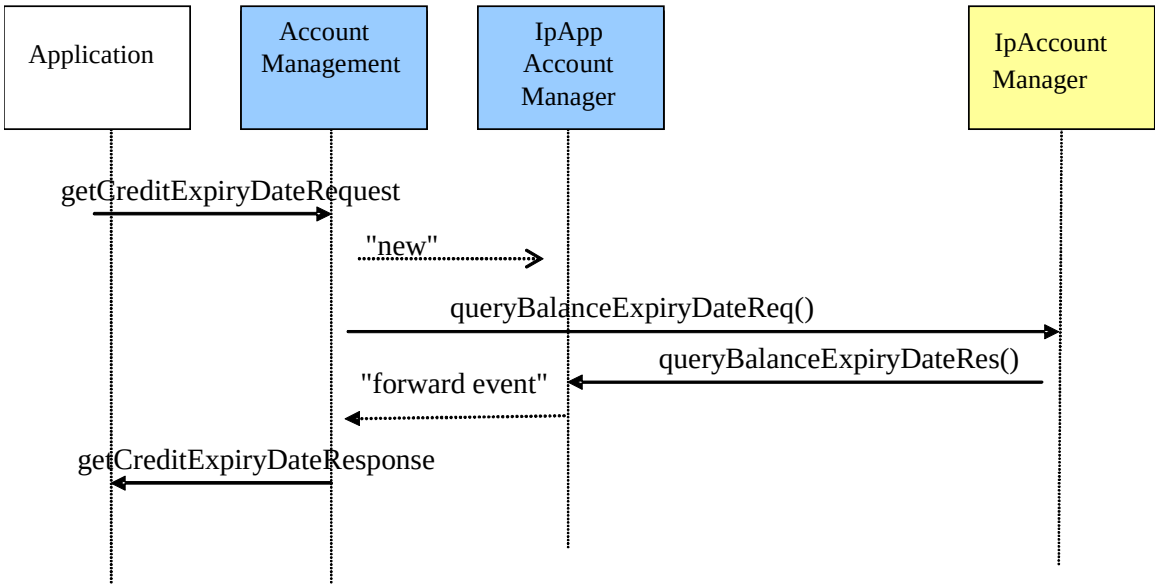


Figure 3

5.4 Increment account balance (Parlay 5.0 only)

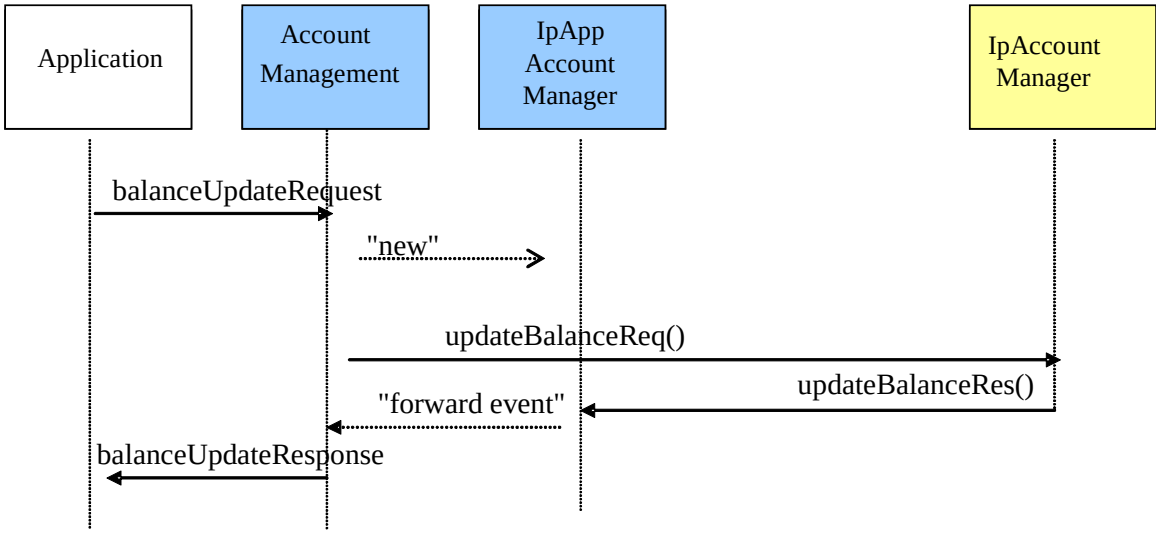


Figure 4

5.5 Increment account balance using Voucher (Parlay 5.0 only)

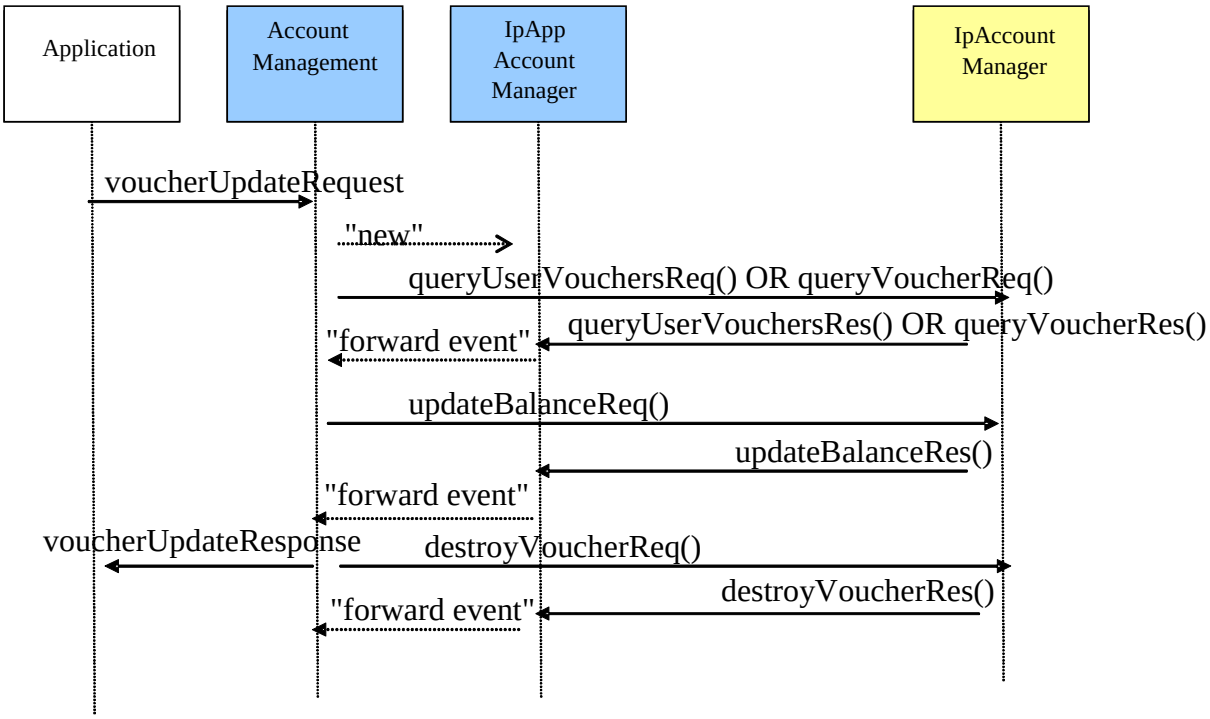


Figure 5

6 Detailed mapping information

6.1 Operations

6.1.1 getBalance

The sequence diagram in clause 5.1 illustrates the flow for the **getBalance** operation.

The **getBalance** operation is synchronous from the Parlay X client's point of view. It is mapped to the following Parlay/OSA methods:

- `IpAccountManager.queryBalanceReq;`
- `IpAppAccountManager.queryBalanceRes;`
- `IpAppAccountManager.queryBalanceErr.`

6.1.1.1 Mapping to `IpAccountManager.queryBalanceReq`

The `IpAccountManager.queryBalanceReq` method is invoked with the following parameters.

Name	Type	Comment
users	TpAddressSet	Populated with one `TpAddress` element, constructed based on the URI provided in the **EndUserIdentifier** part of **getBalance**, mapped as described in TR 102 397-1 [3].

Note that there is no direct mapping to Parlay/OSA for the optional **EndUserPin** part, which contains the end user's credentials for authorizing access to the account. However, there is a mandatory series of authentication and access security steps that are performed between the Account Management Web Service and the Parlay/OSA Framework SCF, for example the Service Agreement Management APIs, that restrict and/or govern access to accounts via the Parlay/OSA Account Management SCF. The credentials supplied to the Parlay X Account Management web service by the Parlay X client (e.g. in the **EndUserPin** part) may be mapped indirectly to a security "access profile" negotiated between the the Account Management web service and the Parlay/OSA Framework and Account Management SCFs.

The result from `IpAccountManager.queryBalanceReq` is used internally to correlate the callbacks: i.e. subsequent invocations of `IpAppAccountManager.queryBalanceRes` or `IpAppAccountManager.queryBalanceErr`.

Parlay exceptions thrown by `IpAccountManager.queryBalanceReq` are mapped to Parlay X exceptions as defined in clause 6.2.4.

6.1.1.2 Mapping from `IpAppAccountManager.queryBalanceRes`

When balance information is available, the `IpAppAccountManager.queryBalanceRes` callback is invoked. It is expected to contain a single `TpBalance` element whose `UserID` field matches the `TpAddress` passed to the balance request operation. The fields of the `TpBalance` element are mapped as follows.

Name	Type	Comment
UserID	TpAddress	Matches the `TpAddress` element passed to the balance request operation.
StatusCode	TpBalanceQueryError	If the value is `P_BALANCE_QUERY_OK`, a result is returned, otherwise an exception is thrown by **getBalance** using the mapping from `TpBalanceQueryError` values to Parlay X Exceptions described in clause 6.2.1.
BalanceInfo	TpBalanceInfo	If the `StatusCode` value (above) is `P_BALANCE_QUERY_OK`, then the **Amount** result returned from **getBalance** will be derived as discussed in the following table.

The following table indicates how Parlay `TpBalanceInfo` data elements are mapped to the **Amount** part:

Name	Type	Comment
Currency	TpString	Not mapped.
ValuePartA	TpInt32	The value of the **Amount** part equals: $(ValuePartA \times 2^{32} + ValuePartB) \times 10^{-Exponent}$
ValuePartB	TpInt32	
Exponent	TpInt32	
AdditionalInfo	TpString	Not mapped.

6.1.1.3 Mapping from `IpAppAccountManager.queryBalanceErr`

If an error prevents the balance information from being reported, the `IpAppAccountManager.queryBalanceErr` callback is invoked. If this occurs, an exception will be thrown by **getBalance** based on the value of the `cause` parameter. Refer to the `TpBalanceQueryError` to Parlay X exception mapping in clause 6.2.1 for details.

6.1.2 getHistory

The sequence diagram in clause 5.2 illustrates the flow for the **getHistory** operation.

The **getHistory** operation is synchronous from the Parlay X client's point of view. It is mapped to the following Parlay/OSA methods:

- `IpAccountManager.retrieveTransactionHistoryReq;`
- `IpAppAccountManager.retrieveTransactionHistoryRes;`
- `IpAppAccountManager.retrieveTransactionHistoryErr.`

6.1.2.1 Mapping to IpAccountManager.retrieveTransactionHistoryReq

The IpAccountManager.retrieveTransactionHistoryReq method is invoked with the following parameters.

Name	Type	Comment
user	TpAddress	The TpAddress element is constructed based on the URI provided in the **EndUserIdentifier** part of **getHistory**, mapped as described in TR 102 397-1 [3].
transactionInterval	TpTimeInterval	The StartTime element is generated from the value of the optional **Date** part. There is no direct mapping to the StopTime element, although the value of the optional **MaxEntries** part may enable the Web Service to derive a coarse estimate for this element of transactionInterval.

Note that there is no direct mapping to Parlay/OSA for the optional **EndUserPin** part, which contains the end user's credentials for authorizing access to the account; however an indirect mapping exists as discussed in clause 6.1.1.1. Note also that there is no direct mapping for the optional **MaxEntries** part.

The result from IpAccountManager.retrieveTransactionHistoryReq is used internally to correlate the callbacks: i.e. subsequent invocations of IpAppAccountManager.retrieveTransactionHistoryRes or IpAppAccountManager.retrieveTransactionHistoryErr.

Parlay exceptions thrown by IpAccountManager.retrieveTransactionHistoryReq are mapped to Parlay X exceptions as defined in clause 6.2.2.

6.1.2.2 Mapping from IpAppAccountManager.retrieveTransactionHistoryRes

When transaction history information is available, the IpAppAccountManager.retrieveTransactionHistoryRes callback is invoked. It is expected to contain a set of TpTransactionHistory elements which are mapped as follows.

Name	Type	Comment
TransactionID	TpAssignmentID	Converted to "string" format, then mapped and included in the **TransactionDetails** part.
TimeStamp	TpDateAndTime	Mapped to the **TransactionDate** part.
AdditionalInfo	TpString	Mapped and included in the **TransactionDetails** part.

6.1.2.3 Mapping from IpAppAccountManager.retrieveTransactionHistoryErr

If an error prevents the balance information from being reported, the IpAppAccountManager.retrieveTransactionHistoryErr callback is invoked. If this occurs, an exception will be thrown by **getHistory** based on the value of the transactionHistoryError parameter. Refer to the TpTransactionHistoryStatus to Parlay X exception mapping in clause 6.2.2 for details.

6.1.3 getCreditExpiryDate (Parlay 5.0 only)

The sequence diagram in clause 5.3 illustrates the flow for the **getCreditExpiryDate** operation.

The **getCreditExpiryDate** operation is synchronous from the Parlay X client's point of view. It is mapped to the following Parlay/OSA methods:

- IpAccountManager.queryBalanceExpiryDateReq;
- IpAppAccountManager.queryBalanceExpiryDateRes;
- IpAppAccountManager.queryBalanceExpiryDateErr.

6.1.3.1 Mapping to IpAccountManager.queryBalanceExpiryDateReq

The IpAccountManager.queryBalanceExpiryDateReq method is invoked with the following parameters.

Name	Type	Comment
users	TpAddressSet	Populated with one TpAddress element, constructed based on the URI provided in the **EndUserIdentifier** part of **getCreditExpiryDate**, mapped as described in TR 102 397-1 [3].

Note that there is no direct mapping to Parlay/OSA for the optional **EndUserPin** part, which contains the end user's credentials for authorizing access to the account; however an indirect mapping exists as discussed in clause 6.1.1.1.

The result from IpAccountManager.queryBalanceExpiryDateReq is used internally to correlate the callbacks: i.e. subsequent invocations of IpAppAccountManager.queryBalanceExpiryDateRes or IpAppAccountManager.queryBalanceExpiryDateErr.

Parlay exceptions thrown by IpAccountManager.queryBalanceExpiryDateReq are mapped to Parlay X exceptions as defined in clause 6.2.4.

6.1.3.2 Mapping from IpAppAccountManager.queryBalanceExpiryDateRes

When balance expiration date information is available, the IpAppAccountManager.queryBalanceExpiryDateRes callback is invoked. It is expected to contain a single TpBalanceExpiryDate element whose UserID field matches the TpAddress passed to the balance expiry date request operation. The fields of the TpBalanceExpiryDate element are mapped as follows.

Name	Type	Comment
UserID	TpAddress	Matches the TpAddress element passed to the balance expiry date request operation.
StatusCode	TpBalanceQueryError	If the value is P_BALANCE_QUERY_OK, a result is returned, otherwise an exception is thrown by **getCreditExpiryDate** using the mapping from TpBalanceQueryError values to Parlay X Exceptions described in clause 6.2.1.
ExpiryDate	TpDateAndTime	If the StatusCode value (above) is P_BALANCE_QUERY _OK, then this field is mapped to the **Date** part of **getCreditExpiryDateResponse.**

6.1.3.3 Mapping from IpAppAccountManager.queryBalanceExpiryDateErr

If an error prevents the balance information from being reported, the IpAppAccountManager.queryBalanceExpiryDateErr callback is invoked. If this occurs, an exception will be thrown by **getCreditExpiryDate** based on the value of the cause parameter. Refer to the TpBalanceQueryError to Parlay X exception mapping in clause 6.2.1 for details.

6.1.4 balanceUpdate (Parlay 5.0 only)

The sequence diagram in clause 5.4 illustrates the flow for the **balanceUpdate** operation.

The **balanceUpdate** operation is synchronous from the Parlay X client's point of view. It is mapped to the following Parlay/OSA methods:

- IpAccountManager.updateBalanceReq;
- IpAppAccountManager.updateBalanceRes;
- IpAppAccountManager.updateBalanceErr.

6.1.4.1 Mapping to IpAccountManager.updateBalanceReq

The IpAccountManager.updateBalanceReq method is invoked with the following parameters.

Name	Type	Comment
user	TpAddress	The TpAddress element is constructed based on the URI provided in the **EndUserIdentifier** part of **balanceUpdate**, mapped as described in TR 102 397-1 [3].
debit	TpBoolean	Set to "True" if the **Amount** part of **balanceUpdateRequest** is negative; otherwise set to "False".
amount	TpBalanceInfo	Set to the absolute (positive) value of the **Amount** part.
period	TpInt32	Set to the value of the **Period** part. May also need to convert from "days" to the time unit used by Parlay/OSA.

Note that there is no mapping to Parlay/OSA for either of the following optional parts:

- **EndUserPin**, which contains the end user's credentials for authorizing access to the account. However an indirect mapping exists as discussed in clause 6.1.1.1.
- **ReferenceCode**, which contains textual information to uniquely identify the request, e.g. in case of disputes.

The result from IpAccountManager.updateBalanceReq is used internally to correlate the callbacks: i.e. subsequent invocations of IpAppAccountManager.updateBalanceRes or IpAppAccountManager.updateBalanceErr.

Parlay exceptions thrown by IpAccountManager.updateBalanceReq are mapped to Parlay X exceptions as defined in clause 6.2.4.

6.1.4.2 Mapping from IpAppAccountManager.updateBalanceRes

If the balance update is successfully performed, the IpAppAccountManager.updateBalanceRes callback is invoked. It contains a single TpBalance parameter containing the current value of the account. This parameter is NOT mapped to the **balanceUpdateResponse** message. The **balanceUpdateResponse** message is invoked.

6.1.4.3 Mapping from IpAppAccountManager.updateBalanceErr

If an error prevents the balance update from being performed, the IpAppAccountManager.updateBalanceErr callback is invoked. If this occurs, an exception will be thrown by **balanceUpdate** based on the value of the cause parameter. Refer to the TpBalanceQueryError to Parlay X exception mapping in clause 6.2.1 for details.

6.1.5 voucherUpdate (Parlay 5.0 only)

The sequence diagram in clause 5.5 illustrates the flow for the **voucherUpdate** operation.

The **voucherUpdate** operation is synchronous from the Parlay X client's point of view. It is mapped to the following Parlay/OSA methods:

- Ip(App)AccountManager.queryUserVouchersReq/Res/Err; OR Ip(App)AccountManager.queryVoucherReq/Res/Err.
- Ip(App)AccountManager.updateBalanceReq/Res/Err.
- IpAccountManager.destroyVoucherReq.

6.1.5.1 Mapping to IpAccountManager.queryUserVouchersReq

The IpAccountManager.queryUserVouchersReq method is invoked with the following parameters.

Name	Type	Comment
user	TpAddress	The TpAddress element is constructed based on the URI provided in the **EndUserIdentifier** part of **voucherUpdate**, mapped as described in TR 102 397-1 [3].

The result from IpAccountManager.queryUserVouchersReq is used internally to correlate the callbacks: i.e. subsequent invocations of IpAppAccountManager.queryUserVouchersRes or IpAppAccountManager.queryUserVouchersErr.

Parlay exceptions thrown by IpAccountManager.queryUserVouchersReq are mapped to Parlay X exceptions as defined in clause 6.2.4.

6.1.5.2 Mapping from IpAppAccountManager.queryUserVouchersRes

When user voucher information is available, the IpAppAccountManager.queryUserVouchersRes callback is invoked. It is expected to contain a set of one or more TpVoucher elements. One of these TpVoucher elements should match the values of the parts of the **voucherUpdateRequest** message, as follows.

Name	Type	Comment
VoucherID	TpAssignmentID	Matches the value of the **VoucherIdentifer** part.
UserID	TpAddress	Matches the value of the **EndUserIdentifier** part.
BalanceInfo	TpBalanceInfo	Not mapped – used during invocation of IpAccountManager.updateBalanceReq (clause 6.1.5.7).

If no TpVoucher element matches these parts of the **voucherUpdateRequest** message, an SVC0002 exception will be thrown by **voucherUpdate**.

6.1.5.3 Mapping from IpAppAccountManager.queryUserVouchersErr

If an error prevents any user voucher information from being reported, the IpAppAccountManager.queryUserVouchersErr callback is invoked. If this occurs, an exception will be thrown by **voucherUpdate** based on the value of the cause parameter. Refer to the TpVoucherError to Parlay X exception mapping in clause 6.2.3 for details.

6.1.5.4 Mapping to IpAccountManager.queryVoucherReq

The IpAccountManager.queryVoucherReq method is invoked with the following parameters.

Name	Type	Comment
voucherID	TpAssignmentID	Matches the value of the **VoucherIdentifer** part.

The result from IpAccountManager.queryVoucherReq is used internally to correlate the callbacks: i.e. subsequent invocations of IpAppAccountManager.queryVoucherRes or IpAppAccountManager.queryVoucherErr.

Parlay exceptions thrown by IpAccountManager.queryVoucherReq are mapped to Parlay X exceptions as defined in clause 6.2.4.

6.1.5.5 Mapping from IpAppAccountManager.queryVoucherRes

When voucher information is available, the IpAppAccountManager.queryVoucherRes callback is invoked. It is expected to contain a single TpVoucher element whose UserID field matches the TpAddress passed to the voucher request operation. The fields of the TpVoucher element are mapped as follows.

Name	Type	Comment
VoucherID	TpAssignmentID	Matches the value of the **VoucherIdentifer** part.
UserID	TpAddress	Matches the value of the **EndUserIdentifier** part.
BalanceInfo	TpBalanceInfo	Not mapped – used during invocation of IpAccountManager.updateBalanceReq (clause 6.1.5.7).

6.1.5.6 Mapping from IpAppAccountManager.queryVoucherErr

If an error prevents the voucher information from being reported, the IpAppAccountManager.queryVoucherErr callback is invoked. If this occurs, an exception will be thrown by v**oucherUpdate** based on the value of the cause parameter. Refer to the TpVoucherError to Parlay X exception mapping in clause 6.2.3 for details.

6.1.5.7 Mapping to IpAccountManager.updateBalanceReq

The IpAccountManager.updateBalanceReq method is invoked with the following parameters.

Name	Type	Comment
user	TpAddress	The TpAddress element is constructed based on the URI provided in the **EndUserIdentifier** part of **voucherUpdate**, mapped as described in TR 102 397-1 [3].
debit	TpBoolean	Set to "False".
amount	TpBalanceInfo	Set to the value of the BalanceInfo field of the TpVoucher element in the IpAppAccountManager.queryVoucherRes callback (clause 6.1.5.5).
period	TpInt32	Set to zero: i.e. no update to the balance expiration period.

The result from IpAccountManager.updateBalanceReq is used internally to correlate the callbacks: i.e. subsequent invocations of IpAppAccountManager.updateBalanceRes or IpAppAccountManager.updateBalanceErr.

Parlay exceptions thrown by IpAccountManager.updateBalanceReq are mapped to Parlay X exceptions as defined in clause 6.2.4.

6.1.5.8 Mapping from IpAppAccountManager.updateBalanceRes

If the balance update is successfully performed, the IpAppAccountManager.updateBalanceRes callback is invoked. It contains a single TpBalance parameter containing the current value of the account. This parameter is NOT mapped to the **voucherUpdateResponse** message. The **voucherUpdateResponse** message is invoked.

6.1.5.9 Mapping from IpAppAccountManager.updateBalanceErr

If an error prevents the balance update from being performed, the IpAppAccountManager.updateBalanceErr callback is invoked. If this occurs, an exception will be thrown by **voucherUpdate** based on the value of the cause parameter. Refer to the TpBalanceQueryError to Parlay X exception mapping in clause 6.2.1 for details.

6.1.5.10 Mapping to IpAccountManager.destroyVoucherReq

The IpAccountManager.destroyVoucherReq method is invoked with the following parameters.

Name	Type	Comment
voucherID	TpAssignmentID	Matches the value of the **VoucherIdentifer** part.

Irrespective of the result of the IpAccountManager.destroyVoucherReq invocation, since the balance update in clause 6.1.5.7 was successful, the voucher must be "designated as used" by the Account Management web service to prevent its reuse in any future invocation of **voucherUpdate.**

6.1.5.11 Unmapped Elements

There is no direct mapping to Parlay/OSA for either of the following optional parts of the **voucherUpdateRequest** message:

- **EndUserPin**, which contains the end user's credentials for authorizing access to the account. However an indirect mapping exists as discussed in clause 6.1.1.1.
- **VoucherPin** part, which contains the voucher's credentials for authentication.
- **ReferenceCode**, which contains textual information to uniquely identify the request, e.g. in case of disputes.

6.2 Exceptions

6.2.1 Mapping from TpBalanceQueryError

The following table indicates how TpBalanceQueryError values are mapped to Parlay X exceptions.

Value	Exception	Notes
P_BALANCE_QUERY _ERROR_UNDEFINED	SVC0001	With error number
P_BALANCE_QUERY _UNKNOWN_SUBSCRIBER	SVC0002	
P_BALANCE_QUERY _UNAUTHORIZED_APPLICATION	SVC0250	
P_BALANCE_QUERY_SYSTEM_FAILURE	SVC0001	With error number

6.2.2 Mapping from TpTransactionHistoryStatus

The following table indicates how TpTransactionHistoryStatus values are mapped to Parlay X exceptions.

Value	Exception	Notes
P_AM_TRANSACTION _ERROR_UNSPECIFIED	SVC0001	With error number
P_AM_TRANSACTION _INVALID_INTERVAL	SVC0002	
P_ AM_TRANSACTION _UNKNOWN_ACCOUNT	SVC0002	
P_ AM_TRANSACTION _UNAUTHORIZED_APPLICATION	SVC0250	
P_ AM_TRANSACTION _PROCESSING_ERROR	SVC0001	With error number
P_ AM_TRANSACTION _SYSTEM_FAILURE	SVC0001	With error number

6.2.3 Mapping from TpVoucherError

The following table indicates how TpVoucherError values are mapped to Parlay X exceptions.

Value	Exception	Notes
P_VOUCHER_ERROR_UNDEFINED	SVC0001	With error number
P_VOUCHER_UNKNOWN_SUBSCRIBER	SVC0002	
P_VOUCHER_UNAUTHORIZED_APPLICATION	SVC0250	
P_VOUCHER_SYSTEM_FAILURE	SVC0001	With error number

6.2.4 Mapping from Parlay/OSA method exceptions

In addition to the common mapping of Parlay/OSA API method exceptions to Parlay X Web Service exceptions, which is defined in TR 102 397-1 [3], there are the following service-specific exception mappings.

Parlay/OSA Exception	Service Exception	Notes
P_UNAUTHORIZED_APPLICATION	SVC0250	
P_INVALID_ASSIGNMENT_ID	SVC0251	With voucher ID. This overrides default mapping in [3] for voucher-related methods.

7 Additional notes

No additional notes.

History

Document history		
V1.1.1	August 2005	Publication