Amdocs Integration with NeuStar

Order Flows

Version 1.0

9/20/2008

# 

# Purpose

This document lays out the message flows for all transactions in a “sunny-day” scenario. No supplements, errors or delays are listed in these flows. Once there is agreement that these flows represent the interaction between NeuStar and Amdocs, the scenarios will be addressed. The purpose of this document is to get high level agreement on the responsibilities of each party.

Also note that the diagrams indicate message flow end-to-end for completeness. Amdocs need only be concerned with the flows between Amdocs and the NeuStar adapter.

# Questions

1. Does Amdocs support the creation of a new LRN in the Amdocs system? This is a fairly rare occurrence and if it is not currently supported, the customer can be supported by either a) opening a ticket with NeuStar or b) using the existing SOA GUI. If Amdocs does support this function now, it can be automated. If not, Neustar suggests an out-of-band procedure. Note: The assignment of an LRN to a ported number is handled in the Portin flows. Additionally, changing the LRN for an already ported number is also supported (see flow below).
2. Similar to #1 above, does Amdocs support the submission of pooling blocks? If
3. Can Amdocs determine whether the number to be ported is wireless or wireline? If so, NeuStar would like Amdocs to populate NPDI and assign different order types for each. If not, the order type will have to be assigned in the NeuStar adapter.

# Assumptions

1. Amdocs will be capable of populating all needed fields in each transaction.
2. The Neustar adapter will validate received messages against the ESROrder schema.
3. The Neustar adapter will ensure that all required fields are populated for the order type/subtype submitted.
4. If either 2 or 3 fail, the order will be NACK’d by the adapter and will not be persisted in the Order DB used by the Workflow Automation layer. Subsequent sends of corrected orders will be submitted as new requests, not supplements.

# ESROrder Schema

The ESROrder schema is a single container that includes all of the requests and responses for the functionality delivered by the WNP platform. Our main focus will be the areas concerned with wireless number porting and stand-alone SOA transactions. The main blocks of information are the following;

* Order Details
* Customer Details
* ICP
* SOA

### Order Details

This block is used in all transactions. It includes the customer submitting the order, the order type being submitted and other pertinent information. On responses, a high level order status is provided. More detailed information can be obtained in the response section corresponding to the product request message.

### Customer Details

Customer Information is used to provide flexibility in meeting complex order management. For example, a wireless port will use the ICP portion of the interface. However, the SOA component of the port requires LRN. Since this is not part of the ICP interface, a node for LRN is provided in the Customer Information.

### ICP

The ICP block is based on the WICIS 4.0 UOM standards. The following messages will be supported for the Amdocs interface.

* createPortRequest
* cancelPortRequest
* changeDueDatePortRequest
* modifyPortRequest
* storeStatus
* validationStatus
* confirmPortResponse
* delayPortResponse
* resolutionRequiredPortResponse

### SOA

The SOA block is based on the Neustar SOA interface and contains request and response messages for all supported SOA transactions. The following messages will be supported for the Amdocs interface.

* SvActivateRequest
* SvDisconnectRequest
* SvModifyRequest
* LrnCreateRequest
* NumberPoolBlockActivateRequest

Note that other requests, such as SvCreateRequest and SvReleaseRequest exist as part of the managed order. As such, they will be used by the Workflow Automation layer but do not need to be exposed to Amdocs. All corresponding responses will be listed in a more detailed API Guide.

# Wireless to Wireless Portin



## Step-by-step walkthrough

|  |  |
| --- | --- |
| # | Description |
| 1 | A wireless port in request is submitted via SOAP to the N\* adapter using the ESROrder order schema. Information will be provided in the OD, CD, and ICP blocks. The OrderType is ‘WIRELESS’ and the OrderSubtype is ‘PORTIN’. The ICP message is createPortRequest. |
| 2 | A synchronous storeStatus is returned by the N\* adapter. |
| 3 | The N\* adapter maps the request to the Workflow Automation (WFA) format using SEA\_REQUEST.xsd. |
| 4 | After validation, WFA sends a WPR to the N\* Clearinghouse (CH). |
| 5 | A synchronous storeStatus is returned by the N\* CH. |
| 6 | The N\* CH sends the WPR to the ONSP or its designee. |
| 7 | A synchronous storeStatus is returned by the ONSP. |
| 8 | After some processing, the ONSP sends an asynchronous validationStatus message to the N\* CH. |
| 9 | A synchronous storeStatus is returned by the N\* CH. |
| 10 | The N\* CH sends the validationStatus message to WFA. |
| 11 | A synchronous storeStatus is returned by WFA. |
| 12 | WFA sends the validationStatus message in the SEA\_RESPONSE.xsd to the N\* adapter. |
| 13 | The N\* adapter maps this response to ESROrder and sends asynchronously to Amdocs. |
| 14 | A synchronous storeStatus is returned by Amdocs. |
| 15 | After processing leads to a confirmation of the WPR request, the ONSP sends an asynchronous confirmPortResponse message to the N\* CH. |
| 16 | A synchronous storeStatus is returned by the N\* CH. |
| 17 | The N\* CH generates an asynchronous validationStatus to the ONSP. |
| 18 | A synchronous storeStatus is returned by the ONSP. |
| 19 | The N\* CH sends the confirmPortResponse message to WFA. |
| 20 | A synchronous storeStatus is returned by WFA. Note that the validationStatus is not required to be returned to the N\* CH. |
| 21 | WFA sends the confirmPortResponse message in the SEA\_RESPONSE.xsd to the N\* adapter. |
| 22 | The N\* adapter maps this response to ESROrder and sends asynchronously to Amdocs. |
| 23 | A synchronous storeStatus is returned by Amdocs. Note that the validationStatus is not required to be returned to the N\* adapter. |
| 24 | At the same time that WFA forwarded the confirmPortResponse message to the N\* adapter, a SOA SvCreateRequest is submitted to the SOA platform in the N\*CH. |
| 25 | WFA asynchronously receives an SvCreateAckNotifcation. |
| 26 | When activation is needed, Amdocs will send a request using ESROrder. Information will be provided in the OD, CD, and SOA blocks. The SOA message is SvActivateRequest. |
| 27 | The N\* adapter maps the request to the Workflow Automation (WFA) format using a schema called SEA\_REQUEST.xsd. |
| 28 | WFA submits an SvActivateRequest to the SOA platform in the N\* CH. |
| 29 | WFA asynchronously receives an SvActivateNotifcation. |
| 30 | A ‘complete’ message is generated and sent to the N\* adapter. |
| 31 | The N\* adapter maps the message to ESROrder and sends to Amdocs. |

# Wireline to Wireless Portin



## Step-by-step walkthrough

|  |  |
| --- | --- |
| # | Description |
| 1 | A wireless port in request is submitted via SOAP to the N\* adapter using the ESROrder order schema. Information will be provided in the OD, CD, and ICP blocks. The OrderType is ‘INTERMODAL’ and the OrderSubtype is ‘PORTIN’. The ICP message is createPortRequest. |
| 2 | A synchronous storeStatus is returned by the N\* adapter. |
| 3 | The N\* adapter maps the request to the Workflow Automation (WFA) format using SEA\_REQUEST.xsd. |
| 4 | WFA maps the ICP data into a CSR Request and sends it to the N\* CH. |
| 5 | The N\* CH sends the CSR to the bonded LEC (ONSP). |
| 6 | The LEC returns a CSR parsed response. This is asynchronous and is usually received in less than 15 seconds. |
| 7 | The N\* CH sends the CSR parsed response to WFA. |
| 8 | A validationStatus is generated and sent to the N\* adapter. |
| 9 | The N\* adapter sends ESROrder containing the validationStatus to Amdocs. |
| 10 | A synchronous storeStatus is returned by the N\* CH. |
| 11 | At the same time that WFA forwarded the validationStatus message to the N\* adapter, an LSR is created from the CSR and submitted to the N\* CH. |
| 12 | The N\* CH sends the LSR to the bonded LEC (ONSP). |
| 13 | The LEC returns a focaccept. |
| 14 | The N\* CH sends the focaccept to WFA. |
| 15 | WFA generates a confirmPortResponse message in the SEA\_RESPONSE.xsd to the N\* adapter. |
| 16 | The N\* adapter maps this response to ESROrder and sends asynchronously to Amdocs. |
| 17 | A synchronous storeStatus is returned by Amdocs. Note that the validationStatus is not required to be returned to the N\* adapter. |
| 18 | At the same time that WFA forwarded the confirmPortResponse message to the N\* adapter, a SOA SvCreateRequest is submitted to the SOA platform in the N\*CH. |
| 19 | WFA asynchronously receives an SvCreateAckNotifcation. |
| 20 | When activation is needed, Amdocs will send a request using ESROrder. Information will be provided in the OD, CD, and SOA blocks. The SOA message is SvActivateRequest. |
| 21 | The N\* adapter maps the request to the Workflow Automation (WFA) format using a schema called SEA\_REQUEST.xsd. |
| 22 | WFA submits an SvActivateRequest to the SOA platform in the N\* CH. |
| 23 | WFA asynchronously receives an SvActivateNotifcation. |
| 24 | A ‘complete’ message is generated and sent to the N\* adapter. |
| 25 | The N\* adapter maps the message to ESROrder and sends to Amdocs. |

# Wireless to Wireless Portout



## Step-by-step walkthrough

|  |  |
| --- | --- |
| # | Description |
| 1 | A Portin request is received by the N\* CH from the NNSP (Or its designee). |
| 2 | A synchronous storeStatus is returned by the N\* CH. |
| 3 | The WPR is sent to Workflow Automation. |
| 4 | A synchronous storeStatus is returned by WFA. |
| 5 | The WPR is forwarded to the N\* adapter where it is mapped into the ESROrder container. |
| 6 | The N\* adapter forwards the ESROrder to Amdocs via a SOAP call. |
| 7 | A synchronous storeStatus is returned by Amdocs. |
| 8 | After some processing, Amdocs sends an asynchronous validationStatus via the ESROrder container. |
| 9 | A synchronous storeStatus is returned by the N\* adapter. |
| 10 | The message is mapped by the N\* adapter to the SEA\_REQUEST container and forwarded to WFA. |
| 11 | The validationStatus message is sent to the N\* CH. |
| 12 | A synchronous storeStatus is returned by the N\* CH. |
| 13 | The validationStatus message is sent to the NNSP. |
| 14 | A synchronous storeStatus is returned by the NNSP. |
| 15 | Amdocs generates a WPRR (confirmPortResponse) message and sends it to the N\* adapter. |
| 16 | A synchronous storeStatus is returned by the N\* CH. |
| 17 | The message is mapped by the N\* adapter to the SEA\_REQUEST container and forwarded to WFA. |
| 18 | The WPRR is sent to the N\* CH. |
| 19 | A synchronous storeStatus is returned by the N\* CH. |
| 20 | The WPRR is sent to the NNSP. |
| 21 | A synchronous storeStatus is returned by the NNSP. |
| 22 | After some processing, the NNSP sends an asynchronous validationStatus to the N\* CH. |
| 23 | A synchronous storeStatus is returned by the N\* CH. |
| 24 | The validationStatus is sent to WFA. |
| 25 | A synchronous storeStatus is returned by WFA. |
| 26 | The validationStatus is forwarded to the N\* adapter where it is mapped into the ESROrder container. |
| 27 | The N\* adapter forwards the ESROrder to Amdocs via a SOAP call. |
| 28 | A synchronous storeStatus is returned by Amdocs. |
| 29 | WFA sends a SOA SvReleaseRequest to the N\* CH. |
| 30 | WFA receives asynchronously a SOA SvReleaseAckNotification. |
| 31 | WFA generates a ‘complete’ transaction and sends it to the N\* adapter. |
| 32 | The N\* adapter forwards the ‘complete’ transaction to Amdocs. |
| 33 | Amdocs sends back an ‘acknowledgement’. |

# SOA Disconnect



## Step-by-step walkthrough

|  |  |
| --- | --- |
| # | Description |
| 1 | A SOA Disconnect request is submitted via SOAP to the N\* adapter using the ESROrder order schema. Information will be provided in the OD, CD, and SOA blocks. The OrderType is ‘SOA’ and the OrderSubtype is ‘DISCONNECT’. The SOA message is SvDisconnectRequest. |
| 2 | The N\* adapter sends back a synchronous ‘acknowledgement’ response. |
| 3 | The adapter maps the request to SEA\_REQUEST and sends it to WFA. |
| 4 | WFA sends the request to the SOA Gateway in the N\* CH. |
| 5 | A synchronous ack is received from SOA. |
| 6 | SOA sends an M-ACTION Request subscriptionVersionDisconnect message to NPAC. |
| 7 | NPAC returns an M-ACTION Response subscriptionVersionDisconnect message to SOA. |
| 8 | An NPACRequestSuccessReply is sent to WFA. |
| 9 | NPAC later sends an asynchronous M-EVENT-REPORT subscriptionVersionStatusAttributeValueChange message to SOA. |
| 10 | SOA sends an SvDisconnectNotification message to WFA. |
| 11 | WFA generates a ‘complete’ transaction and sends it to the N\* adapter. |
| 12 | The N\* adapter forwards the ‘complete’ transaction to Amdocs. |
| 13 | Amdocs sends back an ‘acknowledgement’. |

# SOA LRN Change

## Step-by-step walkthrough

|  |  |
| --- | --- |
| # | Description |
| 1 | A SOA LRN Change request is submitted via SOAP to the N\* adapter using the ESROrder order schema. Information will be provided in the OD, CD, and SOA blocks. The OrderType is ‘SOA’ and the OrderSubtype is ‘LRN CHANGE’. The SOA message is SvDisconnectRequest. |
| 2 | The N\* adapter sends back a synchronous ‘acknowledgement’ response. |
| 3 | The adapter maps the request to SEA\_REQUEST and sends it to WFA. |
| 4 | WFA sends the request to the SOA Gateway in the N\* CH. |
| 5 | A synchronous ack is received from SOA. |
| 6 | SOA sends an M-ACTION Request subscriptionVersionModify message to NPAC. |
| 7 | NPAC returns an M-ACTION Response subscriptionVersionModify message to SOA. |
| 8 | An NPACRequestSuccessReply is sent to WFA. |
| 9 | NPAC later sends an asynchronous M-EVENT-REPORT subscriptionVersionStatusAttributeValueChange message to SOA. |
| 10 | SOA sends an SvChangeNotification message to WFA. |
| 11 | WFA generates a ‘complete’ transaction and sends it to the N\* adapter. |
| 12 | The N\* adapter forwards the ‘complete’ transaction to Amdocs. |
| 13 | Amdocs sends back an ‘acknowledgement’. |

# SOA LRN Create

TBD – Needs to be determined whether this needs automation or its occurrence is so infrequent that interface with SOA Clearinghouse makes sense.

# SOA Number Pooling

TBD – Needs to be determined whether this is supported by Amdocs.