



Connectivity

## Transaction Delivery Agent 4.0

# Functional Overview

This document provides a high level description of the main functional enhancements planned for Transaction Delivery Agent release 4.0, planned to be generally available by end April 2011.

The purpose of this document is to provide an exhaustive description of these enhancements, so that customers can assess how to take advantage of these new features and accordingly plan for it.

September 2010

---

---

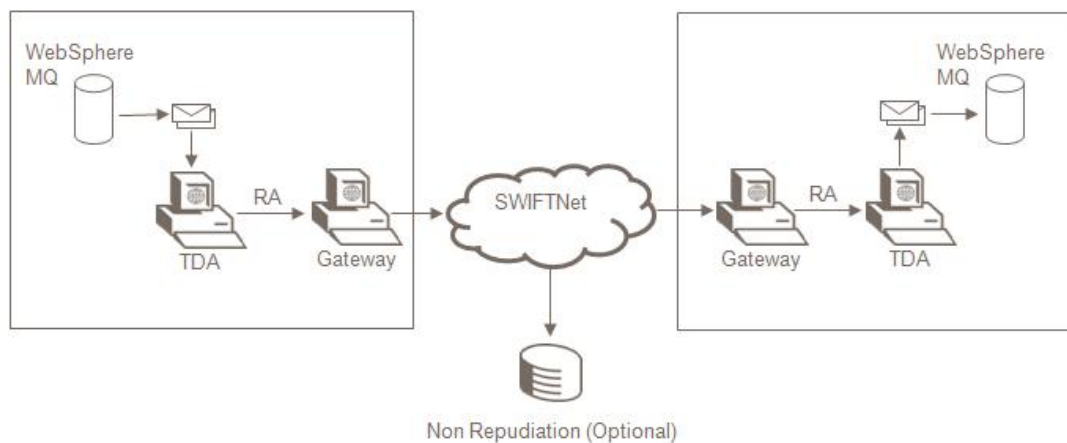
# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
<b>2</b>	<b>Functional Enhancements .....</b>	<b>4</b>
2.1	Forced resynchronisation mechanism .....	4
2.2	End-to-End connectivity tool .....	4
2.3	Dynamic reload of the configuration file.....	4
2.4	Support for multi-services .....	5
2.5	Batching capability .....	5
2.6	Emission and reception service validation .....	5
2.7	Other enhancements .....	6
	<b>Legal Notices .....</b>	<b>8</b>

# 1 Introduction

Transaction Delivery Agent (TDA) provides reliable message transfer with guaranteed delivery between two correspondents. TDA is typically used in small communities where all members use TDA to exchange information with each other.

Transaction Delivery Agent is an Alliance Gateway add-on which makes use of the Remote API Host Adapter (RAHA) and which provides a connection to the institution's applications through the means of the IBM MQ Series middleware interface.



The currently supported releases are 3.0 and 3.1.

TDA 4.0 is planned for end April 2011 and is qualified only for use with Alliance Gateway 7.0 on the following operating systems<sup>1</sup>:

- Windows Server 2008 R2 Standard and Enterprise Edition
- AIX v6.1 TL04 SP3
- Solaris 10 HW 10/09
- IBM WebSphere MQ 7.0.1

TDA 4.0 works with TDA 3.x counterparts, but certain new features will only work once both parties have upgraded to release 4.0.

In line with the SWIFT Release Policy, TDA 3.x. end-of-support date is 31 July 2012.

<sup>1</sup> Pending successful qualification statement

## 2 Functional Enhancements

### 2.1 Forced resynchronisation mechanism

In case of desynchronisation of sequence numbers, TDA 4.0 allows one operator to trigger the automatic resynchronisation of both the emission and the reception TDA instances without having to contact the correspondent's TDA operator to agree on new sequence number to use.

As the re-initialisation of the sequence numbers implies possible message gaps or duplicates, it cannot be activated unless pre-agreed between the two correspondents.

This feature is particularly useful for services for which the message flow must be resumed as soon as possible in case of interruption (business continuity). If the sequencing and once-and-only delivery aspects are important, it must then be guaranteed at the level of the back-office application.

To facilitate handling of duplicate messages by the back-office application, TDA will:

1. Move business messages still present in the emission working queue to the error queue. This is to defer manual processing of possible duplicate messages.
2. Provide an improved logging for use by the back-office application (that is, to assist in the reconciliation of possible duplicate messages).

To benefit from this new functionality, both correspondents must have upgraded to TDA 4.0.

---

**Note** A *ForcedReSynchronise* parameter is introduced in the TDA configuration file at the correspondent level. If set to 'true' for a correspondent, TDA can send and accept forced re-sequence request from their counterparty with which they have this agreement. To send a request, the TDA recovery tool is enhanced with the '-force' option.

---

### 2.2 End-to-End connectivity tool

TDA 4.0 introduces a TDA-to-TDA connectivity tool which allows an operator to check the availability of one or all of his correspondents.

In case the remote TDA instance cannot be reached, the tool returns information to ease the problem investigation, for example, an incorrect provisioning, a configuration error in the local or remote Gateway instance, or the remote TDA which would simply not be started.

Note that this tool does not check elements outside the control of TDA such as the availability of the MQ environment or the back-office application.

This provides an easy way to determine a problem related to TDA vis-à-vis of a problem in the back-office side.

To benefit from this new functionality, both correspondents must have upgraded to TDA 4.0.

---

**Tip** The *tda\_check* tool sends an InterAct message to the specified TDA correspondent and the response provides status information on this correspondent.

---

### 2.3 Dynamic reload of the configuration file

TDA 4.0 allows the operator to change the configuration parameters and reload the entire TDA configuration file while TDA is running.

This allows a flexible approach for customers, or central infrastructure, which need to perform configuration changes without impacting ongoing message flows.

This can for example be used to:

- Enable/disable Gateway connection (Multi-active Gateway configuration).
- “Update” correspondent details (DN updates): this means that messages currently in the emission working queue will be delivered with the newly configured DN.
- “Disable” a correspondent: this in turn will stop corresponding message flow.

SWIFT recommends this feature to experienced users only for use in a tightly controlled environment due to the inherent risks of dynamically changing configuration details during business activity.

---

Tip            The *tdacmd reloadconfig* command loads the new configuration file, parses it for correctness and performs a backup of the previous one. Message processing is then resumed with no impact on the once-and-only once guaranteed delivery paradigm.

---

## 2.4 Support for multi-services

TDA was so far designed for use with one dedicated SWIFTNet service and one request type. As of release 4.0, TDA supports multiple services on a single instance.

This is particularly beneficial to customers willing to re-use their TDA instance for additional SWIFTNet services designed for TDA and who do not want to perform changes in their back-office applications, that is, to provide SWIFTNet related information with each business message.

---

Tip            The configuration parameters that could only be defined in TDA 3.x at the instance level, that is, *AppService*, *AppRequestType*, *RequestSigning*, *ResponseSigning* and *MessageValidation*, can as of release 4.0 be defined per correspondent.

---

## 2.5 Batching capability

For transport efficiency, that is, high throughput of business messages using fewer InterAct messages, TDA 4.0 allows message batching for all three message formats supported (Base 64, XML and FIN).

Batching can be configured for all correspondents of a given service or for a single correspondent.

As batching consists in adding multiple business messages to the same InterAct, it is not compatible with SWIFTNet MVAL service which parses each InterAct message for one single business message.

---

Tip            Batching is performed at MQ queue level with the *Latency* parameter used to define the maximum time elapsed before message emission (whether the maximum InterAct size is reached or not). When batching is used, service name and request type for use in the InterAct envelope are defined in the TDA configuration file.

---

## 2.6 Emission and reception service validation

When performing batching, the service name and request type potentially specified in the business messages are not used to build the InterAct message.

To prevent the creation of a batch containing business messages intended to another service, TDA 4.0 integrates an optional filtering mechanism to validate the service name specified in the message (that is, within <Documentation> tag) against the configured service name for the intended correspondent. Messages that specify other service names shall be moved to the error queue for further investigation. Others shall follow normal TDA processing.

This mechanism is available for both emission and reception.

---

**Tip** If any of the business messages in the batch fails the service name check and if the *StopOnError* parameter is set to False, then the reception TDA moves the erroneous message to the error queue and continues with the processing of the other messages in the batch. Once all messages have been processed successfully, the acknowledgement is sent back. In case *StopOnError* parameter is set to True, the batch is rejected as a whole and the sender TDA is notified.

---

## 2.7 Other enhancements

### ***Silent installation mode***

In line with other Alliance products, TDA 4.0 provides a silent installation framework to remove the need for X-window environments and simplify patch deployments.

### ***TDA running as a background daemon/service***

In line with other Alliance products, TDA 4.0 runs in the background without the need to keep the starting window open anymore.

At start-up time, customer still has all the same possibilities as in TDA 3.x i.e. to create sequence records for the correspondents.

### ***Support the usage of hostnames***

In addition to the support of IP addresses, TDA 4.0 also supports the usage of hostnames in the configuration file to allow dynamic fail-over based on DNS resolution.

### ***Automatic reconnection to MQ queues in case of disconnect***

TDA 4.0 performs automatic reconnection to MQ queues in case of disconnection, that is, due to a network glitch.

### ***SNMP enhancements***

In order to differentiate SNMP events coming from different instances, TDA 4.0 adds to each SNMP trap the source TDA instance name. Additionally, TDA 4.0 now provides the corresponding MIB file which allows efficient integration with monitoring tools.

### ***Log collection***

In line with other Alliance products, TDA 4.0 provides a *tda\_supportinfo* tool that collects all configuration and logs required for problem investigation by Customer Services.

***Support for more correspondents***

TDA 4.0 supports up to 300 correspondents in a single TDA instance.

***Support for MQ servers***

While TDA 3.x was only supported for use with local MQ client software, a separate version of TDA 4.0 is provided for use with a local MQ server.

This is for customers who have specific security policy requirements in place which forbid the usage of MQ client applications.

---

# Legal Notices

## Copyright

SWIFT © 2010. All rights reserved.

You may copy this publication within your organisation. Any such copy must include these legal notices.

## Confidentiality

This publication contains SWIFT or third-party confidential information. Do not disclose this publication outside your organisation without the prior written consent of SWIFT.

## Translations

The English version of SWIFT documentation is the only official and binding version.

## Trademarks

SWIFT is the trade name of S.W.I.F.T. SCRL. The following are registered trademarks of SWIFT: SWIFT, the SWIFT logo, Sibos, SWIFTNet, SWIFTReady, and Accord. Other product, service, or company names in this publication are trade names, trademarks, or registered trademarks of their respective owners.