MarkitWire API: DealSink Cookbook

09 June 2023

1 Introduction

This document is aimed at developers implementing a DealSink API interface to MarkitWire. It covers the technical elements of the trade life-cycle and explains details of the implementation.

1.1 MarkitWire API

MarkitWire is an electronic deal confirmation platform for capturing deals through the generation of same-day legal confirmations and straight-through-processing.

The MarkitWire API supports the complete deal life-cycle from booking to automated confirmation, clearing, regulatory reporting and reconciliation. The system allows deals to be legally confirmed, cleared and reported as part of the deal capture process, removing the need for operations staff to generate paper confirmations. The deal is confirmed on MarkitWire when it reaches the ‘Done’ state. Dealers receive deals directly from brokers connected to the system (when dealing through a broker). When dealing directly, one dealer sends the deal to the other for confirmation. The SWML representation of the deal is considered the ‘confirmation’.

The MarkitWire API also allows broker-dealers, buy-side clients, inter-dealer brokers and prime brokers to confirm, clear, report and reconcile over-the-counter deals directly with all involved, dramatically cutting booking errors, reducing risk and eliminating operational delays. MarkitWire also allows dealers to receive deals from brokers when the counterparty is not connected to MarkitWire (known as single-sided deals). This allows them to take advantage of electronic straight-through-booking regardless of counterparty.

The MarkitWire DealSink API provides MarkitWire participants the ability to capture every event in the trade lifecycle and make unilateral updates to the trade. DealSink API provides the participants with essentially the same functionality that MarkitWire Tracker users have. A MarkitWire Tracker user is a GUI user who can only view trades and cannot submit, affirm, or modify trades. This enables participants to write an interface between their own front, middle or back-office systems and MarkitWire to achieve straight-through processing without having to use MarkitWire Tracker.

2 MarkitWire API General Principles

The MarkitWire API is an event driven (C-callable) API that can be called or wrapped from almost any other language. An example Java wrapper using JNI is also included in the API package. The Java wrapper consists of an additional dynamic library containing JNI interface code and a number of examples demonstrating its use.

2.1 Ownership of Data

Deals submitted are considered to belong to the user ID that logged in to create the session. That is to say those items such as the notifications belong to the user not the session. Deals submitted by one user cannot be seen by any other user that does not belong to the same book.

A MarkitWire API DealSink user account can view all deals entered into MarkitWire by any user belonging to the same organisation with the relevant book ID.

2.2 Blocking and Synchronicity

The MarkitWire API is synchronous. It blocks while performing a task and returns when the task has completed. All MarkitWire API functions are thread safe, but not free threading with the exception of SW_Poll, since MarkitWire version 5.2. Whilst functions can be called with multiple threads, each thread is currently processed sequentially. The aim is to move to a free threading model in the future.

2.3 API Buffers Passed and Returned

Strings passed to the MarkitWire API are taken to be normal null-terminated C strings and should be valid and unchanging for the duration of a specific call to a specific MarkitWire API function. Passing invalid values (except null where documented as permissible) produces undefined behaviour (for example, a protection fault is quite likely). The MarkitWire API calls do not record any string pointers passed, where a value is required to persist the copy of the data internally. As such it is perfectly valid to pass a pointer to a string and then have that pointer become invalid sometime after the MarkitWire API call returns.

Strings returned by the API are similarly normal null-terminated C strings. The user code should not manipulate this buffer, attempt to change any of the bytes, or read past the terminating null. A buffer returned by a particular function on a specific handle is valid until this client code calls SW_ReleaseString on it, at which point it will become invalid.

2.4 Broken Sessions and Resumption Semantics

If a session is broken, existing deals are left in the system (as they belong to the user, not the session). Notifications are not replayed and new notifications are not sent until a new session starts for the same user. As in the rest of the system, a user can only be logged into the system once at any time, so there is no chance of duplicate notifications delivered to multiple applications.

The same user should log in again with a new session to resume activities. When a login is attempted with a user ID that the system thinks is currently logged in, the servers bounce a message from the currently logged in user to check if the connection is dead. If the connection is not dead, the API user's new login will not log out the other logged in instance. The API user will get a message 'APIError: SWERR_LoginLimitReached: "The maximum number of concurrent user log ins the API supports has been reached" '. The GUI Trader Client user's new login will log out the other already logged in API instance.

If a session crashes in a disorderly fashion (i.e. the TCP stack does not close the session as it normally would), a re-login succeeds, but may take longer than normal as the servers check and close down the old session.

While logged in, notifications are delivered, or a synchronisation error is sent. Under certain circumstances failure of server components can lead to a record being saved to the database, but if a server component crashes the corresponding notification is not sent.

3 Getting Started

Before any user is able to use MarkitWire API, a MarkitWire login ID and password are required to be obtained. If you do not have any login credentials, contact your internal MarkitWire System Administrator to get the access approval, and email support@markitserv.com for Production user ID and mserv_uat@markitserv.com for UAT user ID with email subject line ‘MW-NEW USER’.

3.1 Logging into MarkitWire

To connect and log into MarkitWire:

```
        long sessionHandle = -1;
        long loginHandle = -1;
        std::string server(“MyServer”]);
        int timeout = 120;
        int rc;
        if ((rc = SW_Connect(server, timeout, NULL, 
&sessionHandle)) < SWERR        {
            std::cout << "Failed connection (" << 
getError(rc) << ")" << std::endl;
        }
        std::string username(“MyUserName”);
        std::string password((“MyUserPassword”)
        int rc;
        if ((rc = SW_Login(sessionHandle, username, 
password, &loginHandle)) < SWERR_Success)
        {
            std::cout << "Failed to login (" << 
getError(rc) << ")" << std::endl;
            break;
        }
        else
        {
            std::cout << "Successful login: " << 
getError(rc) << std::endl;
        }
```

3.2 Registering a Session State Callback

To register a session state callback: registers a function to call when the state of the server connection changes or an asynchronous internal action fails.

```
// Assuming a valid session notification call back 
function
              rc = 
SW_RegisterSessionStateCallback(SW_SessionStateCallbackPtr 
cfunc);
              if (rc < SWERR_Success)
              {
                    std::cout << "Failed to register 
session state callback" << std::endl;
```

```
                      }
```

3.3 Registering a Deal Notification Callback

To register a deal notification callback: ensures cfunc will be called whenever an event occurs on a user's deals.

```
// Assuming a valid login handle and pointer to a deal
notification call back function
        long retval = SW_RegisterDealNotifyExCallback
(loginHandle, (void*)42, DealNotifyExCallbackPtr cfunc);
        if (retval < SWERR_Success)
        {
            std::cout << "Failed to register notify
callback" << std::endl;
        }
```

3.4 Logging out of MarkitWire

To log out and disconnect from the MarkitWire server:

```
// Assuming previously initialised session and login
handles.
         SW_Logout(loginHandle);
         SW_Disconnect(sessionHandle);
```

3.5 Working with the MarkitWire API

In this section the use of some functions is explained, as well as some tips and features which could be useful for the programmer to know.

3.5.1 Retrying Function Calls

Retry at intervals of not less than three seconds, with the interval doubling between each successive failed retry.

Pseudo code:

```
delay = 3
while ((rc = Api_Function_Call()) != success && delay <
30)
{
    sleep delay * 1000;
    delay = delay * 2;
}
if (rc != success)
{
// A hard error or not one that is going to be resolved
quickly. May need manual intervention.  Stop retrying &
flag/log for someone to look at it...
}
```

3.5.2 Handling Errors

The vast majority of functions in the MarkitWire API return a long value, which is to be interpreted as an error code. If the value returned equals SWERR_Success, the function returned without errors. If a value less than zero is returned, an error occurred. The MarkitWire API application must decide how to handle this.

Errors can be returned due to a number of conditions. As a general design guideline, the MarkitWire API will try to return as specific error codes as possible. Broadly speaking, error codes can be divided in four categories:

Something was wrong with the data provided (for example, SWERR_BadParameter, SW_InvalidXML)

An action was attempted which cannot be performed when the deal is in a certain state (for example, SWERR_ActionUnavailable, SWERR_ActionNotYetAvailable)

The user does not have the right permissions to perform the attempted action (for example, SWERR_NoLegalEntityPerm, SWERR_NoTradeProductPerm)

A system-related error occurred (for example, SWERR_SoftwareMismatch, SWERR_AccountLocked)

For all error messages, it is important to know that the function SW_GetLastErrorSpecificsEx will return more specific information about the error. For example, if an error code SWERR_NoLegalEntityPerm is returned, it means that the MarkitWire API user has not been assigned some legal entities needed to perform the action. This general description would be returned by the function SW_GetErrorDescription. But using SW_GetLastErrorSpecificsEx the user would be told exactly which legal entity for which they are not permissioned. The use of SW_GetLastErrorSpecificsEx is recommended.

3.5.3 Using the “dealVersionHandle”

Deal version handle (or DVH) is a 64 character string uniquely identifying a specific version of a deal. Deal version handles are used to identify the version of a deal to operate on. API functions that perform actions on deals typically take the DVH of the deal, perform the action(s) and then return the DVH of the resulting deal to the caller. In the event that the DVH passed to an API function has been superseded by a later deal version (for example, if the other side of a deal has modified it since the user called the function), then the error SWERR_HandleNotLatest is generally returned. If the DVH passed is invalid (NULL, empty or malformed) it will usually be rejected with SWERR_BadParameter.

For example, if the counterparty of a deal has pulled the deal and affirmed it back with changed details, but for some reason the MarkitWire API user has not yet received a notification. They can try to affirm the deal using the latest DVH they know. The server checks this against the latest DVH available, and determines that they are not the same. The SW_Affirm function would return with the error code SWERR_HandleNotLatest.

If an error SWERR_HandleNotLatest is returned, it means that the DVH which was provided to the function is not the latest. Suggested actions could be to retrieve the state and the latest DVH by using SW_DealGetVersionHandle (or, if more information is needed about the deal, SW_GetActiveDealInfo could be used) and from there decide what action should be taken. If the deal is in a state where the counterparty is supposed to perform an action, the MarkitWire API user needs to do nothing. If the MarkitWire API user is expected to perform an action, he should do so according to the logic.

The DVH changes when data about the deal is stored to the database. This can be either changed deal details, or changed status information. This is the reason that all action functions returns a new DVH which should be used henceforward. The counterparty’s actions will also change the DVH. This is the reason that a DVH is provided with every MarkitWire API notification. To perform an action on the deal, the latest DVH must be provided.

4 Notifications

Notifications are sent on certain events occurring in the deal lifecycle. Notifications always refer to a certain version of a deal. Notifications can be divided into pre-done and post-done notifications.

Pre-done notifications are sent out before a deal has been mutually affirmed by both parties. There is no guarantee that a deal in a pre-done state will actually reach a state of Done. For example, if a deal was sent in error to a party it would be withdrawn. Pre-done notifications are provided for those organisations that wish to update their risk management systems before a deal has been mutually affirmed and released. For those organisations wishing to interface to their internal systems at the point of mutual affirmation (state of Done), or at release, all pre-done notifications can be safely ignored.

Post-done notifications are sent when a deal reaches the mutually affirmed state of Done and when a deal is released. The Done notification is significant since it notifies the MarkitWire API about a legally binding event. On receiving a post-done notification, a MarkitWire API application can safely propagate the information to back office systems.

***Note**:To begin receiving notifications, a MarkitWire API application must first register a callback function to which notifications are sent.*

4.1 Structure of Notifications

Notifications received through the MarkitWire API post-deal processing contain status information about the deal, along with a Deal Id and version information. No economic details of the deal are provided in the notification.

The structure of a pre-done and post-done notification is the same. The state information provided with the notification must be used to determine whether the notification is a pre-done notification or a post-done notification.

With the status information provided, a MarkitWire API interface can determine if it is interested in the full details of the deal or whether the notification can be discarded (for example, if it was a pre-done notification and these are not of interest).

If the interface wants to retrieve more information, for example, economic details about the deal, a separate call must be made to retrieve the corresponding SWML.

4.2 When Notifications are Generated

MarkitWire API notifications will be sent during both broker-to-dealer (B2D) or dealer-to-dealer (D2D) interactions. Below are some examples when notifications will be generated:

- Sending a deal [D2D]
- Picking up a deal [B2D and D2D]
- Rejecting a deal [B2D]
- Accepting a brokered deal [B2D]
- Cancelling a deal (either a broker action or a dealer action) [B2D and D2D]
- Acknowledging a cancel [B2D and D2D]
- Withdrawing a deal (a dealer action) [D2D]

- Mutual affirmation of deal details [B2D and D2D]
- Releasing a deal [B2D and D2D]
- Private data amendment [B2D and D2D]
- Transferring a deal [B2D and D2D]
- Deal is sent to clearing [B2D and D2D]
- Reports are generated [B2D and D2D]
- Reporting detail amendment [D2D]

Private data can be amended through the SW_DealUpate function. In this case, a notification will be sent to the party which performed the unilateral amendment to indicate that the deal has changed.

For the same Deal Id, notifications are expected to arrive in sequential order, but this is not guaranteed. Received notifications should never be assumed to indicate the current state of the deal (it only represents the state of the deal as it was when the notification was sent). This means that if at any point the current state of a deal is needed, a query should be performed to determine this.

4.2.1 Nugatory Updates

Nugatory updates or **CptyNonEconomicUpdate** are “Non Economic” updates that do not require affirmation from the counterparty. Notification of the update will be delivered to both sides of the trade. Certain regulatory fields such as ISIN will not require an affirmation, however MarkitWire is designed to notify both sides about the change.

The update is not a bilateral change as the counterparty does not need to affirm the change, the update is not a unilateral change as both sides see the update.

Updates made to the below fields trigger the **CptyNonEconomicUpdate** notifications in MarkitWire:

- ES UTI
- ES UTI Namespace
- FCA UTI
- FCA UTI Namespace
- Executing Trader ID
- Sales Trader ID
- Desk Location ID
- Broker Location Prefix
- Indirect Cpty Prefix
- Indirect Cpty
- ISIN
- CFI
- Full Name

Here is an example of a notification for a Nugatory update:

```
<Notification>
   <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>8</minorVer>
  <lh>1234</lh>
  <dealId>12345678</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>2</side>
  <privateVer>3</privateVer>
  <newStateStr>CptyNonEconomicUpdate</newStateStr>
</Notification>
```

4.3 Understanding Notifications

When retrieving deal data following a notification, it is important to understand how the retrieved data should be interpreted. This section explains the relevant fields. Below is a sample structure of a notification. Please note that all 64 bit dvh strings will be represented in this document as 'new dvh'.

```
<Notification>
   <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_PendingDirectDealWithMe</newState>
  <minorVer>1</minorVer>
  <lh>1902</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>2</side>
  <privateVer>0</privateVer>
  <newStateStr>Pending</newStateStr>
</Notification>
```

1. Dealer B, under participant Mega Bank Ltd, picks up the deal.

```
Notification received by Dealer B:
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>2</minorVer>
  <lh>1902</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>2</side>
  <privateVer>1</privateVer>
  <newStateStr>PickedUp</newStateStr>
</Notification>
Notification received by Dealer A:
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_AwaitingCptyAction</newState>
  <minorVer>2</minorVer>
  <lh>1901</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>1</privateVer>
  <newStateStr>Pending</newStateStr>
</Notification>
```

2. Dealer B looks at the deal sent through and changes some bilateral data and added private data. He sends the counter-proposal back to Dealer A.

```
Notification received by Dealer B:
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>3</minorVer>
  <lh>1902</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>2</side>
  <privateVer>2</privateVer>
  <newStateStr>Affirm</newStateStr>
</Notification>
Notification received by Dealer A:
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>3</minorVer>
  <lh>1901</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>1</privateVer>
  <newStateStr>CptyAffirm</newStateStr>
</Notification>
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_AwaitingMyAction</newState>
  <minorVer>3</minorVer>
  <lh>1901</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>1</privateVer>
  <newStateStr>Pending</newStateStr>
</Notification>
```

3. Dealer A receives the counter-proposal, and affirms the deal details.

```
Notification received by Dealer A:
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>5</minorVer>
  <lh>1901</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>2</privateVer>
  <newStateStr>Affirm</newStateStr>
</Notification>
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>5</minorVer>
  <lh>1901</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>2</privateVer>
  <newStateStr>Done</newStateStr>
</Notification>
```

Notification received by Dealer B:

```
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_AffirmedDeal</newState>
  <minorVer>6</minorVer>
  <lh>1902</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>2</side>
  <privateVer>3</privateVer>
  <newStateStr>Done</newStateStr>
</Notification>
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>6</minorVer>
  <lh>1902</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>2</side>
  <privateVer>3</privateVer>
  <newStateStr>Done</newStateStr>
</Notification>
```

4. Dealer B changes a few private data details and releases the deal.

Notification received by Dealer B:

```
<Notification>
```

```
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>7</minorVer>
  <lh>1902</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>2</side>
  <privateVer>4</privateVer>
  <newStateStr>Released</newStateStr>
</Notification>
```

5. Dealer A releases the deal.

```
Notification received by Dealer A:
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>8</minorVer>
  <lh>1901</lh>
  <dealId>30144547</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>3</privateVer>
  <newStateStr>Released</newStateStr>
</Notification>
```

4.4 After Reaching an Affirmed State, Deal is Transferred before Release

1. Dealer A, working for participant X, has agreed all deal details with Dealer B, working for participant Y.

2. Dealer B releases the deal.

```
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>6</minorVer>
  <lh>1902</lh>
  <dealId>30144868</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>2</side>
  <privateVer>3</privateVer>
  <newStateStr>Released</newStateStr>
</Notification>
```

3. Dealer A transfers the deal to Dealer C who also works for participant X.

Notification received by Dealer A:

```
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>7</minorVer>
  <lh>1901</lh>
  <dealId>30144868</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>2</privateVer>
  <newStateStr>Transferred</newStateStr>
</Notification>
```

Notification received by Dealer C:

```
<Notification>
  <contractState>New</contractState>
```

```
    <dvh>new dvh</dvh>
    <newState>SWDS_DealUpdated</newState>
    <minorVer>1</minorVer>
    <lh>1903</lh>
    <dealId>30144868</dealId>
    <tradeAttrFlags>0</tradeAttrFlags>
    <brokerId/>
    <majorVer>1</majorVer>
    <productType>IRS</productType>
    <prevDVH/>
    <side>1</side>
    <privateVer>1</privateVer>
    <newStateStr>Sent</newStateStr>
  </Notification>
  <Notification>
    <contractState>New</contractState>
    <dvh>new dvh</dvh>
    <newState>SWDS_DealUpdated</newState>
    <minorVer>5</minorVer>
    <lh>1903</lh>
    <dealId>30144868</dealId>
    <tradeAttrFlags>0</tradeAttrFlags>
    <brokerId/>
    <majorVer>1</majorVer>
    <productType>IRS</productType>
    <prevDVH/>
    <side>1</side>
    <privateVer>2</privateVer>
    <newStateStr>Done</newStateStr>
  </Notification>
  <Notification>
    <contractState>New</contractState>
    <dvh>new dvh</dvh>
    <newState>SWDS_IncomingTransfer</newState>
    <minorVer>7</minorVer>
    <lh>1903</lh>
    <dealId>30144868</dealId>
    <tradeAttrFlags>0</tradeAttrFlags>
    <brokerId/>
    <majorVer>1</majorVer>
    <productType>IRS</productType>
    <prevDVH/>
    <side>1</side>
    <privateVer>2</privateVer>
    <newStateStr>Done</newStateStr>
  </Notification>
  <Notification>
    <contractState>New</contractState>
    <dvh>new dvh</dvh>
    <newState>SWDS_DealUpdated</newState>
    <minorVer>7</minorVer>
    <lh>1903</lh>
    <dealId>30144868</dealId>
    <tradeAttrFlags>0</tradeAttrFlags>
    <brokerId/>
    <majorVer>1</majorVer>
    <productType>IRS</productType>
    <prevDVH/>
    <side>1</side>
    <privateVer>2</privateVer>
    <newStateStr>Transferred</newStateStr>
  </Notification>
```

4. Dealer C picks up the transfer and changes some private data details and releases the deal.

Notification received by Dealer C:

```
  <Notification>
    <contractState>New</contractState>
    <dvh>new dvh</dvh>
    <newState>SWDS_DealUpdated</newState>
    <minorVer>8</minorVer>
    <lh>1903</lh>
    <dealId>30144868</dealId>
    <tradeAttrFlags>0</tradeAttrFlags>
    <brokerId/>
    <majorVer>1</majorVer>
    <productType>IRS</productType>
    <prevDVH/>
```

```
          <side>1</side>
          <privateVer>2</privateVer>
          <newStateStr>TransferPickUp</newStateStr>
        </Notification>
        <Notification>
          <contractState>New</contractState>
          <dvh>new dvh</dvh>
          <newState>SWDS_DealUpdated</newState>
          <minorVer>9</minorVer>
          <lh>1903</lh>
          <dealId>30144868</dealId>
          <tradeAttrFlags>0</tradeAttrFlags>
          <brokerId/>
          <majorVer>1</majorVer>
          <productType>IRS</productType>
          <prevDVH/>
          <side>1</side>
          <privateVer>3</privateVer>
          <newStateStr>Released</newStateStr>
        </Notification>
```

Notification received by Dealer A:

```
        <Notification>
          <contractState>New</contractState>
          <dvh>new dvh</dvh>
          <newState>SWDS_DealUpdated</newState>
          <minorVer>8</minorVer>
          <lh>1901</lh>
          <dealId>30144868</dealId>
          <tradeAttrFlags>0</tradeAttrFlags>
          <brokerId/>
          <majorVer>1</majorVer>
          <productType>IRS</productType>
          <prevDVH/>
          <side>1</side>
          <privateVer>2</privateVer>
          <newStateStr>TransferPickUp</newStateStr>
        </Notification>
        <Notification>
```

```
      <contractState>New</contractState>
      <dvh>new dvh</dvh>
      <newState>SWDS_DealUpdated</newState>
      <minorVer>9</minorVer>
      <lh>1901</lh>
      <dealId>30144868</dealId>
      <tradeAttrFlags>0</tradeAttrFlags>
      <brokerId/>
      <majorVer>1</majorVer>
      <productType>IRS</productType>
      <prevDVH/>
      <side>1</side>
      <privateVer>3</privateVer>
      <newStateStr>Released</newStateStr>
    </Notification>
```

Note:*Transferring a deal will not change the state the deal was in before it was transferred. The private data will not change when the deal is transferred, all details will be the same as they were for Dealer A*

4.5 Sending a New Deal and the Counterparty Withdraws

1. Dealer A, working for participant X, sends a deal to Dealer B within participant Y.

Notification received by Dealer A:

```
    <Notification>
      <contractState>New</contractState>
      <dvh>new dvh</dvh>
      <newState>SWDS_DealUpdated</newState>
      <minorVer>1</minorVer>
      <lh>1901</lh>
      <dealId>30159102</dealId>
      <tradeAttrFlags>0</tradeAttrFlags>
      <brokerId/>
      <majorVer>1</majorVer>
      <productType>IRS</productType>
      <prevDVH/>
      <side>1</side>
      <privateVer>1</privateVer>
      <newStateStr>Sent</newStateStr>
```

```
        </Notification>
```

Notification received by Dealer B:

```
        <Notification>
          <contractState>New</contractState>
          <dvh>new dvh</dvh>
          <newState>SWDS_PendingDirectDealWithMe</newState>
          <minorVer>1</minorVer>
          <lh>1902</lh>
          <dealId>30159102</dealId>
          <tradeAttrFlags>0</tradeAttrFlags>
          <brokerId/>
          <majorVer>1</majorVer>
          <productType>IRS</productType>
          <prevDVH/>
          <side>2</side>
          <privateVer>0</privateVer>
          <newStateStr>Pending</newStateStr>
        </Notification>
```

2. Dealer B, working for participant Y, picks up the deal.

Notification received by Dealer B:

```
        <Notification>
          <contractState>New</contractState>
          <dvh>new dvh</dvh>
          <newState>SWDS_DealUpdated</newState>
          <minorVer>2</minorVer>
          <lh>1902</lh>
          <dealId>30159102</dealId>
          <tradeAttrFlags>0</tradeAttrFlags>
          <brokerId/>
          <majorVer>1</majorVer>
          <productType>IRS</productType>
          <prevDVH/>
          <side>2</side>
          <privateVer>1</privateVer>
          <newStateStr>PickedUp</newStateStr>
        </Notification>
```

Notification received by Dealer A:

```
        <Notification>
```

```
    <contractState>New</contractState>
    <dvh>new dvh</dvh>
    <newState>SWDS_AwaitingCptyAction</newState>
    <minorVer>2</minorVer>
    <lh>1901</lh>
    <dealId>30159102</dealId>
    <tradeAttrFlags>0</tradeAttrFlags>
    <brokerId/>
    <majorVer>1</majorVer>
    <productType>IRS</productType>
    <prevDVH/>
    <side>1</side>
    <privateVer>1</privateVer>
    <newStateStr>Pending</newStateStr>
  </Notification>
```

3. Dealer B looks at the deal and does not agree with the details at all. He withdraws.

Notification received by Dealer B:

```
  <Notification>
    <contractState>New</contractState>
    <dvh>new dvh</dvh>
    <newState>SWDS_DealUpdated</newState>
    <minorVer>4</minorVer>
    <lh>1902</lh>
    <dealId>30159102</dealId>
    <tradeAttrFlags>0</tradeAttrFlags>
    <brokerId/>
    <majorVer>1</majorVer>
    <productType>IRS</productType>
    <prevDVH/>
    <side>2</side>
    <privateVer>2</privateVer>
    <newStateStr>Withdrawn</newStateStr>
  </Notification>
```

Notification received by Dealer A:

```
  <Notification>
    <contractState>New</contractState>
    <dvh>new dvh</dvh>
    <newState>SWDS_AwaitingAcknowledgement</newState>
    <minorVer>4</minorVer>
    <lh>1901</lh>
    <dealId>30159102</dealId>
    <tradeAttrFlags>0</tradeAttrFlags>
    <brokerId/>
    <majorVer>1</majorVer>
    <productType>IRS</productType>
    <prevDVH/>
    <side>1</side>
    <privateVer>2</privateVer>
    <newStateStr>Withdrawn</newStateStr>
  </Notification>
  <Notification>
    <contractState>New</contractState>
    <dvh>new dvh</dvh>
    <newState>SWDS_DealUpdated</newState>
    <minorVer>4</minorVer>
    <lh>1901</lh>
    <dealId>30159102</dealId>
    <tradeAttrFlags>0</tradeAttrFlags>
    <brokerId/>
    <majorVer>1</majorVer>
    <productType>IRS</productType>
    <prevDVH/>
    <side>1</side>
    <privateVer>2</privateVer>
    <newStateStr>Cancelled</newStateStr>
  </Notification>
```

4. Dealer A acknowledges that the trade was withdrawn by the counterparty.

```
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_nud</newState>
  <minorVer>5</minorVer>
  <lh>1901</lh>
  <dealId>30159102</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>3</privateVer>
  <newStateStr>CancelAcknowledged</newStateStr>
</Notification>
```

Also note the following:

- The withdrawal could follow several iterations of counter-proposals having been sent between the dealers. If so, that would not cause any additional notifications to be sent to the MarkitWire API post-deal processing.
- Participant X sees newState = “Cancelled” since the counterparty withdrew the deal. Participant Y sees newState = “Withdrawn” since it withdrew the deal.
- Participant X gets another notification with newState = “CancelAcknowledged” when Dealer A acknowledges that the trade has been withdrawn by the counterparty.

4.6 After Sending a New Deal the Sender Withdraws before Picking up

1. Dealer A, working for participant X, sends a deal to Dealer B within participant Y.

Notification received by Dealer A:

```
<Notification>
  <contractState>New</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>1</minorVer>
  <lh>1901</lh>
  <dealId>30159809</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>1</privateVer>
  <newStateStr>Sent</newStateStr>
</Notification>
```

Notification received by Dealer B:

```
<Notification>
  <contractState>New</contractState>
```

```
          <dvh>new dvh</dvh>
          <newState>SWDS_PendingDirectDealWithMe</newState>
          <minorVer>1</minorVer>
          <lh>1902</lh>
          <dealId>30159809</dealId>
          <tradeAttrFlags>0</tradeAttrFlags>
          <brokerId/>
          <majorVer>1</majorVer>
          <productType>IRS</productType>
          <prevDVH/>
          <side>2</side>
          <privateVer>0</privateVer>
          <newStateStr>Pending</newStateStr>
        </Notification>
```

2. Dealer A withdraws the deal.

Notification received by Dealer A:

```
        <Notification>
          <contractState>New</contractState>
          <dvh>new dvh</dvh>
          <newState>SWDS_DealUpdated</newState>
          <minorVer>2</minorVer>
          <lh>1901</lh>
          <dealId>30159809</dealId>
          <tradeAttrFlags>0</tradeAttrFlags>
          <brokerId/>
          <majorVer>1</majorVer>
          <productType>IRS</productType>
          <prevDVH/>
          <side>1</side>
          <privateVer>2</privateVer>
          <newStateStr>Withdrawn</newStateStr>
        </Notification>
```

Notification received by Dealer B:

```
        <Notification>
          <contractState>New</contractState>
          <dvh>new dvh</dvh>
          <newState>SWDS_Terminated</newState>
          <minorVer>2</minorVer>
```

```
  <lh>1902</lh>
  <dealId>30159809</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>2</side>
  <privateVer>0</privateVer>
  <newStateStr>Withdrawn</newStateStr>
</Notification>
```

5 Brokered Deal Notifications

The examples in this section illustrate some common broker-to-dealer (brokered deal) scenarios to show when notifications are generated and what information they contain. (Note: The notification examples display the difference from the previous notification for each side except for the first notification for a deal id.)

5.1 Both Dealers Accept, Affirm and Release the Deal

Event	Giga Broker	AAA Bank	Mega Bank
Giga Broker submits a deal to AAA Bank and Mega Bank.	<Notification> <contractState/> <dvh>new dvh</dvh> <newState>SWDS_Active</newState> <minorVer>1</minorVer> <lh>1901</lh> <dealId>30673032</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VVTT</brokerId> <majorVer>1</majorVer> <productType>Single Currency Interest Rate Swap</productType> <prevDVH/> <side>-1</side> <privateVer>0</privateVer> <newStateStr>Active</newStateStr> </Notification>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1902</lh> <dealId>30673032</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VVTT</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1903</lh> <dealId>30673032</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VVTT</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>3</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>
AAA Bank picks up the deal.	<Notification> <dvh>new dvh</dvh> </Notification>	<Notification> <dvh>new dvh</dvh>	

		<newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>	
Mega Bank picks up the deal.	<Notification> <dvh>new dvh</dvh> </Notification>		<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>3</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>
AAA Bank accepts/affirms the deal.	<Notification> <dvh>new dvh</dvh> </Notification>	<Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>1</privateVer> <newStateStr>Accepted</newStateStr> </Notification>	
Mega Bank accepts/affirms the deal.	<Notification> <dvh>new dvh</dvh> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_Accepted</newState>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_AwaitingCptyAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>5</minorVer> <newStateStr>Accepted</newStateStr> </Notification>

	<newStateStr>Accepted</newStateStr> </Notification>	<Notification> <newState>SWDS_DealUpdated</newState> <newStateStr>AcceptedWithThirdParty</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_AffirmedDeal</newState> <minorVer>9</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <newStateStr>Done</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_AwaitingCptyAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <newStateStr>AcceptedWithThirdParty</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_AffirmedDeal</newState> <minorVer>8</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh>

			<newState>SWDS_DealUpdated</newState> <newStateStr>Done</newStateStr> </Notification>
AAA Bank releases the deal.		<Notification> <dvh>new dvh</dvh> <minorVer>10</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>	
Mega Bank releases the deal.			<Notification> <dvh>new dvh</dvh> <minorVer>11</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>

5.2 I Accept but my Counterparty Rejects; Broker Resends the Deal

Event	Giga Broker	AAA Bank	Mega Bank
Giga Broker submits a deal to AAA Bank and Mega Bank.	<Notification> <contractState/> <dvh>new dvh</dvh> <newState>SWDS_Active</newState> <minorVer>1</minorVer> <lh>1901</lh> <dealId>30688016</dealId>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1902</lh>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1903</lh>

	<tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT1</brokerId> <majorVer>1</majorVer> <productType>Single Currency Interest Rate Swap</productType> <prevDVH/> <side>-1</side> <privateVer>0</privateVer> <newStateStr>Active</newStateStr> </Notification>	<dealId>30688016</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT1</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>	<dealId>30688016</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT1</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>3</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>
AAA Bank picks up and accepts the deal.	<Notification> <dvh>new dvh</dvh> </Notification> <Notification> <dvh>new dvh</dvh> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>3</minorVer> <newStateStr>Accepted</newStateStr> </Notification>	
Mega Bank rejects the deal without picking it up.	<Notification> <dvh>new dvh</dvh>	<Notification> <dvh>new dvh</dvh>	

	<newState>SWDS_Withdrawn</newState> <newStateStr>Withdrawn</newStateStr> </Notification>	<newState>SWDS_AwaitingBrokerAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr> </Notification>	
Giga Broker resends the deal.	<Notification> <dvh>new dvh</dvh> <newState>SWDS_Active</newState> <dealId>30688167</dealId> <brokerId>9999999980VTVT2</brokerId> <newStateStr>Active</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <dealId>30688167</dealId> <brokerId>9999999980VTVT2</brokerId> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <dealId>30688167</dealId> <brokerId>9999999980VTVT2</brokerId> </Notification>
AAA Bank picks up, accepts/affirms the deal.	<Notification> <dvh>new dvh</dvh> </Notification> <Notification> <dvh>new dvh</dvh> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>3</minorVer>	

		<newStateStr>Accepted</newStateStr> </Notification>	
Mega Bank picks up the deal.	<Notification> <dvh>new dvh</dvh> </Notification>		<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>4</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>
Mega Bank accepts/affirms the deal.	<Notification> <dvh>new dvh</dvh> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_Accepted</newState> <newStateStr>Accepted</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_AwaitingCptyAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr> </Notification> <Notification> <newState>SWDS_DealUpdated</newState> <newStateStr>AcceptedWithThirdParty</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_AffirmedDeal</newState> <minorVer>9</minorVer>	<Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <newStateStr>Accepted</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_AwaitingCptyAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <newStateStr>AcceptedWithThirdParty</newStateStr>

		<privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <newState>SWDS_DealUpdated</newState> </Notification>	</Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_AffirmedDeal</newState> <minorVer>8</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <newState>SWDS_DealUpdated</newState> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <newStateStr>Done</newStateStr> </Notification>

5.3 I Accept, Counterparty Rejects; Broker Resends Deal Changes Recipient

Event	Giga Broker	AAA Bank	Mega Bank	Markit Bank
Giga Broker submits a deal to AAA Bank and Mega Bank.	<Notification> <contractState/> <dvh>new dvh</dvh> <newState>SWDS_Active</newState> <minorVer>1</minorVer> <lh>1901</lh>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState>	

	<dealId>30688440</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT3</brokerId> <majorVer>1</majorVer> <productType>Single Currency Interest Rate Swap</productType> <prevDVH/> <side>-1</side> <privateVer>0</privateVer> <newStateStr>Active</newStateStr> </Notification>	<minorVer>1</minorVer> <lh>1902</lh> <dealId>30688440</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT3</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>	<minorVer>1</minorVer> <lh>1903</lh> <dealId>30688440</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT3</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>3</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>	
AAA Bank picks up and accepts the deal.	<Notification> <dvh>new dvh</dvh> <newState>SWDS_Active</newState> </Notification> <Notification> <dvh>new dvh</dvh> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification> <Notification>		

		<dvh>new dvh</dvh> <minorVer>3</minorVer> <newStateStr>Accepted</newStateStr> </Notification>		
Mega Bank rejects the deal without picking it up.	<Notification> <dvh>new dvh</dvh> <newState>SWDS_Withdrawn</newState> <newStateStr>Withdrawn</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_AwaitingBrokerAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr> </Notification>		
Giga Broker resends the deal to AAA and Markit Bank.	<Notification> <contractState/> <dvh>new dvh</dvh> <newState>SWDS_Active</newState> <minorVer>1</minorVer> <lh>1901</lh> <dealId>30688500</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT4</brokerId>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1902</lh> <dealId>30688500</dealId> <tradeAttrFlags>0</tradeAttrFlags>		<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1904</lh> <dealId>30688500</dealId> <tradeAttrFlags>0</tradeAttrFlags>

	<majorVer>1</majorVer> <productType>Single Currency Interest Rate Swap</productType> <prevDVH/> <side>-1</side> <privateVer>0</privateVer> <newStateStr>Active</newStateStr> </Notification>	<brokerId>9999999980VTVT4</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>		<brokerId>9999999980VTVT4</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>3</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>
AAA Bank picks up, accepts/affirms the deal.	<Notification> <dvh>new dvh</dvh> </Notification> <Notification> <dvh>new dvh</dvh> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>3</minorVer> <newStateStr>Accepted</newStateStr> </Notification>		

Markit Bank picks up the deal.	<Notification> <dvh>new dvh</dvh> </Notification>			<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>4</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>
Markit Bank accepts/affirms the deal.	<Notification> <dvh>new dvh</dvh> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_Accepted</newState> <newStateStr>Accepted</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_AwaitingCptyAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr> </Notification> <Notification> <newState>SWDS_DealUpdated</newState> <newStateStr>AcceptedWithThirdParty</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh>		<Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <newStateStr>Accepted</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_AwaitingCptyAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr> </Notification> <Notification> <newState>SWDS_DealUpdated</newState>

		<newState>SWDS_AffirmedDeal</newState> <minorVer>9</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <newState>SWDS_DealUpdated</newState> </Notification>		<newStateStr>AcceptedWithThirdParty</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_AffirmedDeal</newState> <minorVer>8</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <newState>SWDS_DealUpdated</newState> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <newStateStr>Done</newStateStr> </Notification>

5.4 I Accept, Counterparty Rejects; Broker Resends to Different Addressee

Event	Giga Broker	AAA Bank (aaa trader)	AAA Bank (aaa trader 1)	Mega Bank
Giga Broker submits a deal to	<Notification> <contractState/>	<Notification>	<Notification>	<Notification> <contractState>

Mega Bank and AAA Bank (aaa trader and aaa trader 1).	<dvh>new dvh</dvh> <newState>SWDS_Active</newState> <minorVer>1</minorVer> <lh>1901</lh> <dealId>30723570</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT5</brokerId> <majorVer>1</majorVer> <productType>Single Currency Interest Rate Swap</productType> <prevDVH/> <side>-1</side> <privateVer>0</privateVer> <newStateStr>Active</newStateStr> </Notification>	<contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1902</lh> <dealId>30723570</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT5</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>	<contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1903</lh> <dealId>30723570</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT5</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>	New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1904</lh> <dealId>30723570</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT5</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>3</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>
AAA Bank (aaa trader) picks up the deal.	<Notification> <dvh>new dvh</dvh> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_Terminated</newState>	

		<minorVer>2</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>	<minorVer>2</minorVer> <privateVer>1</privateVer> </Notification> <Notification> <newState>SWDS_DealUpdated</newState> <newStateStr>PickedUp</newStateStr> </Notification>	
AAA Bank (aaa trader) accepts the deal.	<Notification> <dvh>new dvh</dvh> </Notification>	<Notification> <dvh>new dvh</dvh> <minorVer>3</minorVer> <newStateStr>Accepted</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <minorVer>3</minorVer> <newStateStr>Accepted</newStateStr> </Notification>	
Mega Bank picks up the deal, but rejects the deal.	<Notification> <dvh>new dvh</dvh> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_Withdrawn</newState> <newStateStr>Withdrawn</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_AwaitingBrokerAction</newState> <minorVer>7</minorVer> <newStateStr>Pending</newStateStr> </Notification>		<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>4</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification> <Notification>

				<dvh>new dvh</dvh> <minorVer>7</minorVer> <privateVer>2</privateVer> <newStateStr>Rejected</newStateStr> </Notification>
Giga Broker resends the deal to Mega Bank, and AAA Bank (aaa trader 1). (aaa trader is not included.)	<Notification> <contractState/> <dvh>new dvh</dvh> <newState>SWDS_Active</newState> <minorVer>1</minorVer> <lh>1901</lh> <dealId>30723641</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT6</brokerId> <majorVer>1</majorVer> <productType>Single Currency Interest Rate Swap</productType> <prevDVH/> <side>-1</side>		<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1903</lh> <dealId>30723641</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT6</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_PendingBrokeredDeal</newState> <minorVer>1</minorVer> <lh>1904</lh> <dealId>30723641</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT6</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/>

	<privateVer>0</privateVer> <newStateStr>Active</newStateStr> </Notification>		<privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>	<side>3</side> <privateVer>0</privateVer> <newStateStr>Pending</newStateStr> </Notification>
AAA Bank aaa trader 1 picks up the deal, and accepts/affirms the deal.	<Notification> <dvh>new dvh</dvh> </Notification> <Notification> <dvh>new dvh</dvh> </Notification>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1902</lh> <dealId>30723641</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId>9999999980VTVT6</brokerId> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>1</privateVer>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>3</minorVer> <newStateStr>Accepted</newStateStr> </Notification>	

		<newStateStr>PickedUp</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>3</minorVer> <newStateStr>Accepted</newStateStr> </Notification>		
Mega Bank picks up the deal.	<Notification> <dvh>new dvh</dvh> </Notification>			<Notification> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>4</minorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>
Mega Bank accepts/affirms the deal.	<Notification> <dvh>new dvh</dvh> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_Accepted</newState>	<Notification> <dvh>new dvh</dvh> <minorVer>6</minorVer> <newStateStr>AcceptedWithThirdParty</newStateStr> </Notification> <Notification>	<Notification> <dvh>new dvh</dvh> <newState>SWDS_AwaitingCptyAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr>	<Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <newStateStr>Accepted</newStateStr> </Notification> <Notification>

	<newStateStr>Accepted</newStateStr> </Notification>	<dvh>new dvh</dvh> <minorVer>9</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification>	</Notification> <Notification> <newState>SWDS_DealUpdated</newState> <newStateStr>AcceptedWithThirdParty</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_AffirmedDeal</newState> <minorVer>9</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <newState>SWDS_DealUpdated</newState> </Notification>	<dvh>new dvh</dvh> <newState>SWDS_AwaitingCptyAction</newState> <minorVer>6</minorVer> <newStateStr>Pending</newStateStr> </Notification> <Notification> <newState>SWDS_DealUpdated</newState> <newStateStr>AcceptedWithThirdParty</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newState>SWDS_AffirmedDeal</newState> <minorVer>8</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <newState>SWDS_

				DealUpdated</newState> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <newStateStr>Done</newStateStr> </Notification>
Mega Bank releases the deal.				<Notification> <dvh>new dvh</dvh> <minorVer>10</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>
AAA Bank (aaa trader 1) releases the deal.		<Notification> <dvh>new dvh</dvh> <minorVer>11</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <minorVer>11</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>	

6 Private Data and Deal Amendment Notifications

6.1 Private Data

Private data is unique to each side of the deal. In this section, private data is explained to a level sufficient to understand private data retrieval and manipulation using the MarkitWire API. The private data can be divided into deal-related data and status information.

Private Data XML is passed to all functions where an action on the deal can lead to private data being saved, such as SW_DealPickup, SW_DealAffirm, SW_SubmitPostTradeEvent, etc. It is also used for unilateral updates on deals using SW_DealUpdate Sink Update. Sink Update XML is passed to the SW_DealUpdate function to indicate which private fields of a deal should be updated. When changing private data, the swPrivateDataVersionId value is automatically updated by the API function. If a Private Data field value is to be left unchanged, this field should be omitted from the Private Data update altogether. Passing empty elements is used to remove existing value.

Private Data is viewable in SWML in swPrivateData container and retrievable by calling SW_DealGetSWML.

The most commonly used Private Data fields are listed in below sections as examples. For more Private Data fields and information, please refer to MarkitWire API Documentation – Private Data XML and Sink Update XML sections.

6.1.1 Deal-Related Data Fields

Sample deal-related private data fields:

- Book Id (swTradingBookId)
- Internal Trade Id (swPrivateTradeId)
- Broker Party Id (swBrokerPartyId)
- Brokerage Currency (swBrokerageAmount/currency)
- Brokerage Amount (swBrokerageAmount/amount)
- Sales Credit (swSalesCredit)
- Additional Fields 1-80 (swAdditionalField)
- Broker (swBrokerPartyId)
- Broker Trade Id (swBrokerTradeId)
- Offsetting Trade Id (swOffsettingTradeId Note: settable by the broker submission)
- Compression Type (swCompressionType)
- Execution Method (swExecutionMethod)
- Clearing Broker Id (swClearingBrokerId)
- Private Fund Id (swPrivateFundId)

6.1.2 Status Information Fields

Sample status-related private data fields:

- Private Booking State (swPrivateBookingState)
- Private Process State (swPrivateProcessState)
- Private Deal Sink Comment (swPrivateDealSinkComment)
- Private DealSink Confirmation (swPrivateDealSinkConfirmation)
- Private Data Version Id (swPrivateDataVersionId)
- Send for Clearing (swSendForClearing)

The swPrivateProcessState contains information about which state the deal is in. It can take one of the following values:

- Pending
- Withdrawn
- Done
- Released

Pending means it is in a pre-done state for that particular Contract Version.

*Note: There could have been previous **Contract Versions** which have reached an affirmed state. Withdrawn means that this **Contract Version** has been withdrawn. Any new actions (for example, amendment, cancellation) would lead to a new **Contract Version** and associated private data. Done means the deal has been mutual affirmed by both parties. Released means that this side of the deal has been released.*

Only the MarkitWire system can change the swPrivateProcessState parameter. Changes to this parameter will follow user or broker actions performed in the system.

The swPrivateBookingState is a free-text field that can be set through the MarkitWire API post-deal processing or through the Tracker user via GUI. It can be used to support a trade flow where Middle Office staff change the booking state to reflect its progress through their particular internal workflow.

The swPrivateDealSinkComment is a free-text field that can be used for holding any additional information pertinent to the current swPrivateBookingState value. For example, if the swPrivateBookingState indicated an error state, this field could be used to hold error message text that would be visible to a user within the Tracker application. This element is not mandatory, and will only appear in the SWML if it is assigned a value.

The swPrivateDealSinkConfirmation field is a free-text field that can be used by a MarkitWire API application in a number of different ways. Its intended purpose is to allow a MarkitWire API application to write back a value into the field confirming that a specific deal version has been acknowledged/received/processed etc. This field can then be used when performing a start-up or recovery query to identify deals that have not yet been processed.

6.2 Unilaterally Amending Private Data

Dealer A, working for participant X, retrieves a deal, which is at pending direct with counterparty Dealer B, for unilateral amend. Dealer A makes some private data updates:

```
<SinkUpdate>
  <swPrivateData>
    <swPrivateTradeId>A983365</swPrivateTradeId>
    <swTradingBookId>BRYAN</swTradingBookId>
  </swPrivateData>
</SinkUpdate>
        <Notification>
          <contractState>New</contractState>
          <dvh>new dvh</dvh>
          <newState>SWDS_DealUpdated</newState>
          <minorVer>2</minorVer>
          <lh>1901</lh>
          <dealId>30671858</dealId>
          <tradeAttrFlags>0</tradeAttrFlags>
          <brokerId/>
          <majorVer>1</majorVer>
          <productType>IRS</productType>
          <prevDVH/>
          <side>1</side>
          <privateVer>2</privateVer>
          <newStateStr>Pending</newStateStr>
        </Notification>
```

Note*: When private data is changed, the private version will increase by one.*

7 Allocations Notifications

The examples in this section illustrate allocation scenarios between Party A and B, the notifications generated and the information they contain. (Note: The notification examples display the difference from the previous notification for each side except for the first notification for a deal id.)

Event	Party A	Party B
Party A : Trade alleged	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>1</minorVer> <lh>1901</lh> <dealId>3474986</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer> <newStateStr>Sent</newStateStr> </Notification>	
Party B : Trade Pick up		<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1902</lh> <dealId>3474986</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side>

		<privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>
Party B : Trade Affirm	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <minorVer>5</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification>	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newStateStr>Done</newStateStr> </Notification>
Party A : Trade Release	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <minorVer>7</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>	
Party B : Trade Release		<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <minorVer>6</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>
Party A : Allocations : Created, affirmed and Released	<Notification> <contractState>Allocated</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>1</minorVer>	

```
  <lh>1901</lh>
  <dealId>3474986</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>2</majorVer>
  <productType>IRS</productType>
  <prevDVH/>
  <side>1</side>
  <privateVer>1</privateVer>
  <newStateStr>Sent</newStateStr>
</Notification>
<Notification>
<dvh>new dvh</dvh>
  <minorVer>5</minorVer>
  <privateVer>2</privateVer>
  <newStateStr>Done</newStateStr>
</Notification>
<Notification>
<contractState>Allocated</contractState>
<dvh>new dvh</dvh>
  <minorVer>6</minorVer>
  <privateVer>3</privateVer>
  <newStateStr>Released</newStateStr>
</Notification>
<Notification>
  <contractState>New-Allocation</contractState>
  <dvh>new dvh</dvh>
  <newState>SWDS_DealUpdated</newState>
  <minorVer>2</minorVer>
  <lh>1901</lh>
  <dealId>3474987</dealId>
  <tradeAttrFlags>0</tradeAttrFlags>
  <brokerId/>
  <majorVer>1</majorVer>
  <productType>IRS</productType>
```

	<prevDVH/> <side>2</side> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification> <Notification> <contractState>New-Allocation</contractState> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>7</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification> <Notification> <contractState>New-Allocation</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1901</lh> <dealId>3474988</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType>	

	<prevDVH/> <side>2</side> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>7</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>	
Party B : Allocations : Created, affirmed and Released		<Notification> <contractState>Allocated</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1902</lh> <dealId>3474986</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>2</majorVer> <productType>IRS</productType> <prevDVH/>

		<side>2</side> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>7</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification> <Notification> <contractState>New-Allocation</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>1</minorVer> <lh>1902</lh> <dealId>3474987</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer>

		<newStateStr>Sent</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>6</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification> <Notification> <contractState>New-Allocation</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>1</minorVer> <lh>1902</lh> <dealId>3474988</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer> <newStateStr>Sent</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh>

		<minorVer>6</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>

8 Novation Notifications

8.1 Novation Process Explained

Novation can be submitted by either one of the original trade's two parties. The content of each notification is party specific and provides information which can be used to identify which trades the notification refers to:

- The Outgoing party is able to see that a novation notification refers to a Novated trade and an associated novation fee.
- The Remaining party is able to see that the novation notification refers to a New-Novated trade and the associated Novated trade.
- The Incoming party is able to see that the novation notification refers to a New-Novated trade and an associated novation fee.

When a notification for a New-Novated trade is received by the Remaining party the contract state for the associated Novated trade are updated.

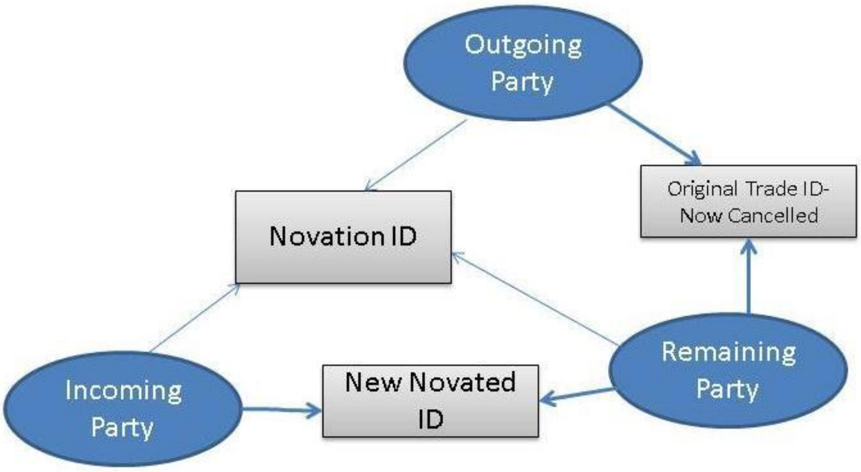
As the novation process results in the Novated contract state for the original trade, the booking states for the Novated and New-Novated trades are kept in synch, such that they are the same until the New-Novated trade is released.

No economic details for novation will be provided in the notifications. Full trade details for a New-Novated or Novated trade, including the novation confirmation details and any associated novation fee are available via a function call `SW_DealGetSWML` to retrieve the SWML.

In the novation container in SWML, eventId is the Novation ID. The Novation ID is the common trade id for all three counterparties involved in the novation, and is used to link the individual trades within a novation process.

The trade id associated with the New-Novated trade only appears for the sides which have a New-Novated trade associated with the Novation Confirmation (for example, Remaining and Incoming party).

The trade id associated with the Novated trade only appears for the sides which have a Novated trade associated with the Novation Confirmation (for example, Outgoing and Remaining party).



The original trade with Novated contract state will behave like cancelled for full novation. For partial novation, the original trade with Novated-Partial contract state will remain.

8.2 Private Booking State and Book ID Kept in Sync across Novated Deals Post-Release

If either the Booking State or Book Id is changed in the Novated or New-Novated deal then the change is carried over to the other deal and the private version is updated on both deals. If any other private data is changed (and the State and Book ID are left unchanged) no data is duplicated across and the private version of the unchanged deal is left untouched.

8.3 Novation Scenario

The following table explains which notifications will be received by each participant for a Novation. Not all participants involved in the deal will receive a notification for every action, so only the parties that receive a notification are indicated against each step. (Note: The notification examples display the difference from the previous notification for each side except for the first notification for a deal id.)

Event	Party A (Outgoing OP)	Party B (Remaining RP)	Party C (Incoming IP)
Party A : Trade alleged	<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState> SWDS_DealUpdated</newState> <minorVer>1</minorVer> <lh>1901</lh> <dealId>3473437</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer> <newStateStr>Sent</newStateStr> </Notification>		

Party B : Trade Pick up		<Notification> <contractState>New</contractState> <dvh>new dvh</dvh> <newState> SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1902</lh> <dealId>3473437</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>1</privateVer> <newStateStr> PickedUp</newStateStr> </Notification>	
Party B : Trade Affirm	<Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <newStateStr>Done</newStateStr> </Notification>	

Party A : Trade Release	<Notification> <dvh>new dvh</dvh> <minorVer>6</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>		
Party B : Trade Release		<Notification> <dvh>new dvh</dvh> <minorVer>7</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>	
Party A : Outgoing Party - Novation Alleged	<Notification> <contractState>Novated</contractState> <dvh>new dvh</dvh> <minorVer>1</minorVer> <majorVer>2</majorVer> <privateVer>1</privateVer> <newStateStr>Sent</newStateStr> </Notification>		
Party B : Remaining Party - Novation Pick up		<Notification> <contractState>Novated</contractState> <dvh>new dvh</dvh> <minorVer>2</minorVer> <majorVer>2</majorVer> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>	

		<Notification> <contractState>New-Novated</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1902</lh> <dealId>3473440</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>	
Party B : Remaining Party - Novation Affirmed		<Notification> <contractState>Novated</contractState> <dvh>new dvh</dvh> <minorVer>3</minorVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <contractState>New-Novated</contractState> <dvh>new dvh</dvh> <minorVer>3</minorVer>	

		<newStateStr>Affirm</newStateStr> </Notification>	
Party C - Incoming Party - New-Novated Trade - Pickup			<Notification> <contractState>New-Novated</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>4</minorVer> <lh>1903</lh> <dealId>3473440</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>
Party C - Incoming Party - New-Novated Trade - Affirmed	<Notification> <dvh>new dvh</dvh> <minorVer>6</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <newStateStr>CptyAffirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>6</minorVer> <newStateStr>CptyAffirm</newStateStr>	<Notification> <dvh>new dvh</dvh> <minorVer>6</minorVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>9</minorVer>

		</Notification> <Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <privateVer>2</privateVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>8</minorVer> <privateVer>2</privateVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>8</minorVer> <newStateStr>Done</newStateStr> </Notification>	<newStateStr>Done</newStateStr> </Notification>
Party A - Outgoing Party - Novation Release	<Notification> <dvh>new dvh</dvh> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>		
Party B - Remaining Party - Release		<Notification> <dvh>new dvh</dvh> <minorVer>8</minorVer>	

		<privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>10</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>	
Party C - Incoming Party - Release			<Notification> <dvh>new dvh</dvh> <minorVer>11</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>

9 Prime Brokerage Notifications

Executing Broker sends trade to a participant and the participant's Prime Broker; all parties agree to all the trade details. The following are sample notifications received for prime brokerage trade scenario. (Note: The notification examples display the difference from the previous notification for each side except for the first notification for a deal id.)

Event	Party A (Executing Broker)	Party B	Party C (Prime Broker)
Party A(Executing Broker) : Trade alleged	<Notification> <contractState>PrimeBrokered</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>1</minorVer> <lh>1901</lh> <dealId>3474975</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer> <newStateStr>Sent</newStateStr> </Notification>		
Party B : Trade Pick up		<Notification> <contractState>PrimeBrokered</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer>	

		<lh>1902</lh> <dealId>3474975</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>	
Party C (Prime Broker) : Deal Pickup			<Notification> <contractState>PrimeBrokered</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>3</minorVer> <lh>1903</lh> <dealId>3474975</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>3</side>

			<privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>
Party B : Deal Affirm	<Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <privateVer>1</privateVer> <newStateStr>PrimaryAgree</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <privateVer>1</privateVer> <newStateStr>PrimaryAgree</newStateStr> </Notification>	<Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <privateVer>1</privateVer> <newStateStr>PrimaryAgree</newStateStr> </Notification>
Party C (Prime Broker) : Deal Accept	<Notification> <dvh>new dvh</dvh> <minorVer>7</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>10</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification> <Notification> <contractState>New-PrimeBrokered</contractState> <dvh>new dvh</dvh>	<Notification> <dvh>new dvh</dvh> <minorVer>8</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>11</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification> <Notification> <contractState>New-PrimeBrokered</contractState> <dvh>new dvh</dvh>	<Notification> <dvh>new dvh</dvh> <minorVer>9</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>12</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification> <Notification> <contractState>New-PrimeBrokered</contractState> <dvh>new dvh</dvh>

	<newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1901</lh> <dealId>3474976</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>	<newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1902</lh> <dealId>3474977</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>1</privateVer> <newStateStr>PickedUp</newStateStr> </Notification>	<newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1903</lh> <dealId>3474976</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer> <newStateStr>Sent</newStateStr> </Notification> <Notification> <contractState>New-PrimeBrokered</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1903</lh> <dealId>3474977</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer>

			<productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer> <newStateStr>Sent</newStateStr> </Notification>
Trade Affirm by Prime Broker - Party C (Trade between Party A & C)	<Notification> <dvh>new dvh</dvh> <minorVer>5</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification>		<Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Affirm</newStateStr> </Notification> <Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification>
Trade Affirm by Prime Broker - Party C (Trade between Party B & C)			<Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Affirm</newStateStr> </Notification>

			<Notification> <dvh>new dvh</dvh> <minorVer>4</minorVer> <privateVer>2</privateVer> <newStateStr>Done</newStateStr> </Notification>
Party A - Deal Release	<Notification> <dvh>new dvh</dvh> <minorVer>6</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>		
Party B - Deal Release		<Notification> <dvh>new dvh</dvh> <minorVer>6</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>	
Party C – Deal Release between Party B & C			<Notification> <dvh>new dvh</dvh> <minorVer>7</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>

Party C – Deal Release between Party A & C			<Notification> <dvh>new dvh</dvh> <minorVer>7</minorVer> <privateVer>3</privateVer> <newStateStr>Released</newStateStr> </Notification>

10 Inter-Dealer Clearing

MarkitWire DealSink can be used to send deals to the CCP for clearing and receive clearing status update, reject or accept clearing notifications. The following sections provide more information on the clearing function calls and notifications.

10.1 Bank-Specific Function Calls

It is possible for a bank to check via a function call whether a deal is eligible for clearing (`SW_DealCheckEligibility` and passing in parameter *eligibilityType* "clearing") and to retrieve the reason(s) why a deal is ineligible for clearing in XML format (returned by `SW_DealCheckEligibility`).

A bank can set the Send for Clearing private data by using the `SW_DealUpdate` function call and passing Sink Update XML:

```
<SinkUpdate>
        <swPrivateData>
                <swSendForClearing>1</swSendForClearing>
        </swPrivateData>
</SinkUpdate>
```

`swSendForClearing` flag can take in a string value of "true", "1", "yes"; or "false", "0", "no". If a deal is not eligible for clearing, a specific error code (`SWERR_DealNotEligibleToClear`) will be returned to indicate this error and the returned clearing status XML will contain the details about the ineligible reason.

10.2 DealSink Clearing Notifications

A notification is generated for each side to the deal on clearing initiation (when the new *Clearing* contract version is created). Since the CCP appears on both sides of the deal they receive two separate notifications with distinct side IDs that are updated individually.

Note that it is enough for CCP to signal back clearing acceptance on one side of the trade. It does not matter which side is used, accepting one side will accept the trade as a whole (i.e. both sides).

Below are notification samples demonstrating status update from CCP, reject clear and accept clear from CCP:

Event	**AAA Bank**	**Mega Bank**	**CCP - LCH Demo** (CCP side is included for reference only)

AAA Bank and Mega Bank sent the deal to clearing respectively, and CCP received both deals sent to clearing notifications.	`<Notification>`
	`<contractState>New</contractState>`
	`<dvh>new dvh</dvh>`
	`<newState>SWDS_DealUpdated</newState>`
		`<Notification>`
		`<contractState>New</contractState>`
		`<dvh>new dvh</dvh>`
		`<newState>SWDS_DealUpdated</newState>`
			After AAA Bank sent the deal to clearing using SW_DealUpdate:
			`<Notification>`
			`<contractState>Clearing</contractState>`
			`<dvh>new dvh</dvh>`

	<minorVer>8</minorVer> <lh>1904</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>4</privateVer> <newStateStr>Released</newStateStr> </Notification> <Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>1</minorVer> <lh>1904</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/>	<minorVer>9</minorVer> <lh>1902</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>1</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>4</privateVer> <newStateStr>Released</newStateStr> </Notification> <Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1902</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/>	<newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1903</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>2</majorVer> <productType>IRS</productType> <prevDVH/> <side>3</side> <privateVer>0</privateVer> <newStateStr>SentForClearing</newStateStr> </Notification> **After Mega Bank sent the deal to clearing using SW_DealUpdate:** <Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1903</lh>

	<majorVer>2</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer> <newStateStr>SentForClearing</newStateStr> </Notification>	<majorVer>2</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>1</privateVer> <newStateStr>SentForClearing</newStateStr> </Notification>	<dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>2</majorVer> <productType>IRS</productType> <prevDVH/> <side>4</side> <privateVer>0</privateVer> <newStateStr>SentForClearing</newStateStr> </Notification>
CCP Status Update, AAA Bank and Mega Bank received status update notifications.	<Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>3</minorVer> <lh>1904</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>2</majorVer>	<Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>4</minorVer> <lh>1902</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>2</majorVer>	**CCP sent below status update to both AAA Bank and Mega Bank using SW_DealUpdate:** <SCML> <StatusUpdate> <ClearingServicePartyATradeReference>MW30653644-A</ClearingServicePartyATradeReference> <ClearingServicePartyBTradeReference>MW30653644-B</ClearingServicePartyBTradeReference> <ClearingServicePartyAStatusUpdate>Pending action</ClearingServicePartyAStatusUpdate> <ClearingServicePartyBStatusUpdate>Pending

	<productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>2</privateVer> <newStateStr>UpdatedForClearing</newStateStr> </Notification>	<productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>2</privateVer> <newStateStr>UpdatedForClearing</newStateStr> </Notification>	action</ClearingServicePartyBStatusUpdate> </StatusUpdate> <MatchingService>SWIRE</MatchingService> <MatchingServiceTradeReference>30653644</MatchingServiceTradeReference> </SCML>
CCP Reject Clear, AAA Bank and Mega Bank received Reject Clear notifications.	<Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>7</minorVer> <lh>1904</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>2</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>3</privateVer> <newStateStr>Rejected	<Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>8</minorVer> <lh>1902</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>2</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>3</privateVer> <newStateStr>Rejected	**CCP sent below reject clear reason using SW_DealUpdate:** <SCML> <Reject> <RejectReason> <ReasonCode>007</ReasonCode> <ReasonText>Invalid tenor</ReasonText> </RejectReason> </Reject> <MatchingService>SWIRE</MatchingService> <MatchingServiceTradeReference>30653644</MatchingServiceTradeReference> </SCML>

	ForClearing</newStateStr> </Notification>	ForClearing</newStateStr> </Notification>	
AAA Bank resent the deal to clearing, CCP received the notifications.	**AAA sent the deal to clearing again:** <Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>1</minorVer> <lh>1904</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>3</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>1</privateVer> <newStateStr>SentForClearing</newStateStr> </Notification>	**After CCP rejection, only one side needs to resend the deal for clearing. Mega side received deal sent to clearing notification after AAA sent the deal to clearing:** <Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1902</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>3</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>1</privateVer> <newStateStr>SentForClearing</newStateStr> </Notification>	**CCP received both deals sent to clearing notifications:** **AAA side:** <Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1903</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>3</majorVer> <productType>IRS</productType> <prevDVH/> <side>3</side> <privateVer>0</privateVer> <newStateStr>SentForClearing</newStateStr> </Notification> Mega side: <Notification>

			<contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>2</minorVer> <lh>1903</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>3</majorVer> <productType>IRS</productType> <prevDVH/> <side>4</side> <privateVer>0</privateVer> <newStateStr>SentForClearing</newStateStr> </Notification>
CCP did Accept Clear; AAA Bank and Mega Bank received Accept Clear notifications.	<Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>7</minorVer>	<Notification> <contractState>Clearing</contractState> <dvh>new dvh</dvh> <newState>SWDS_DealUpdated</newState> <minorVer>8</minorVer>	**CCP accepted clearing:** <SCML> <Accept> <ClearingServicePartyATradeReference>MW30653644-A</ClearingServicePartyATradeReference> <ClearingServicePartyBTradeReference>MW306

	<lh>1904</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>3</majorVer> <productType>IRS</productType> <prevDVH/> <side>1</side> <privateVer>3</privateVer> <newStateStr>RegisteredForClearing</newStateStr> </Notification>	<lh>1902</lh> <dealId>30653644</dealId> <tradeAttrFlags>0</tradeAttrFlags> <brokerId/> <majorVer>3</majorVer> <productType>IRS</productType> <prevDVH/> <side>2</side> <privateVer>3</privateVer> <newStateStr>RegisteredForClearing</newStateStr> </Notification>	53644-B</ClearingServicePartyBTradeReference> <AcceptText>Accept clear</AcceptText> </Accept> <MatchingService>SWIRE</MatchingService> <MatchingServiceTradeReference>30653644</MatchingServiceTradeReference> </SCML>

11 Recovery

Recovery is a critical part of a client's implementation because MarkitWire does not replay notifications for events that occurred when the client was not connected to MarkitWire.

Recovery and backlog processing involves steps for processing changes to deals that occurred while the client application was not connected to MarkitWire. It ensures that the client's own systems are in sync with MarkitWire whenever a connection is established.

All state changes that occurred while the client was not connected to MarkitWire should be recovered as part of the backlog/recovery processing, these should be processed before notifications generated by the deal call-back are processed.

When performing a recovery query, the MarkitWire API interface will need to query for the status of deals matching specified search criteria, and use this information to determine if further deal details are needed or not.

In order to query for deals, the SW_QueryDeals Sink Query function should be used. This function allows a MarkitWire API interface to query for deals based on parameters such as below:

- Activity Date/Time Range
- Private Process State
- Private Booking State
- Private Deal Sink Confirmation
- Deal Id or range of Deal Ids
- Private Trade Id
- Deal Sink Confirmation
- Activity Type

The exact format of the query is specified in the MarkitWire API Documentation. There is no default value for the Private Process State if no private data is specified in the query.

If the Deal Id is used (named tradeId in the query), it is possible to retrieve all versions of the deal through use of the AllVersions element in the query.

***Note**: When retrieving all versions, any contract versions of a deal that were withdrawn before ever being mutually affirmed (Done) will also appear in the results set. These can be easily identified by looking at the private process state or private booking state which will contain the value "Withdrawn" for any version of the deal that was withdrawn.*

If the Deal Id is used, but AllVersions is set to "0" or omitted, only the latest non-Withdrawn version of the deal will be returned. In other words, the latest contract version with a Private Process State set equal to a value other than 'Withdrawn'.

***Note**: A query with private process state set to 'Released', with no private booking state criteria specified, will only return those deals in a 'Released' state with a private booking state of 'Released', i.e. not changed from its default value. In order to retrieve 'Released' deals which have the private booking state set (for example, set to 'Received', 'Validated', 'Cleared', 'Error'* *or another user defined value) it is necessary to specify the **swPrivateBookingState** element in the query and set it to an appropriate value.*

In the query, the swPrivateDealSinkConfirmation element can take a string value, or one of the reserved values “0” or “1” as described in the MarkitWire API Documentation. Searches for a specified string value are case-sensitive.

When calling SW_QueryDeals, an optional callback function can be provided. If it is provided, one function call containing all deals matching the specified criteria will be made to the QueryHandler callback function specified in the call to SW_QueryDeals. If no callback function is provided, the function will return synchronously with the result set.

For each deal matching the original query, the following information will be returned in the Sink Query results. Users can select to turn on ExtendedInfo if more information is needed. Please refer to the MarkitWire API Documentation for more details.

- tradeId
- ContractVersion swTradeVersionId
- ContractState swTradeStatus
- Side
- Private Data Version Id swPrivateDataVersionId
- Private Process State swPrivateProcessState
- Private Booking State swPrivateBookingState

With this information the MarkitWire API interface can use the SW_DealGetSWML function to get the SWML details for the deals where this is needed and use this information to update relevant risk management and back-office systems.

SW_QueryDeals also supports the DealInfo XML as a parameter to return Deal State XML, however this is not recommended for a Dealsink implementation as deals which are not in a released state will not be returned. This is recommended for a Dealer API implementation as it returns information pertinent to a Dealer API implementation such as the deal state.

Appendix A: Additional Documentation

MarkitWire has a number of other documents that should be read in conjunction with this document. These are available for download from the MarkitWire secure website and include the following.

MarkitWire API Reference

The following MarkitWire API reference documents are also available:

MarkitWire API Documentation

This document is available as index.html file in apidocs folder in each API download. It describes Release Notes, Supported Platforms, XML Reference, Thin API Networking Guide, Tracing, API functions modules, etc. in the form of html pages.

MarkitWire API Integration Good Practice

This document is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire API Integration Good Practice'. It provides a comprehensive guide for integrating with the MarkitWire API.

MarkitWire Business Processes and Flows

For in-depth descriptions of the MarkitWire business processes and the relevant process flow diagrams, see the following documents:

MarkitWire Business Processes & Trade Life-Cycle

This document provides a comprehensive guide to the functionality available on the MarkitWire platform either by using MarkitWire Trader, MarkitWire Trader (Web) or the MarkitWire API.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire Business Processes and Trade Life-Cycle'.

Business Process Flows: Direct and Brokered Deals

This document describes the different activity process flows for direct deals (dealer to dealer and dealer to end user) and brokered deals (broker to dealer and broker to end user) on the MarkitWire system.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire Process Flows - Direct and Brokered Deals'.

Business Process Flows - Novations

This document describes the different activity process flows for novated deals on the MarkitWire system.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire Process Flows – Novations'.

Business Process Flows - Prime Brokerage

This document describes the different activity process flows for prime brokered deals on the MarkitWire system.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire Process Flows - Prime Brokerage'.

MarkitWire Process Flows - Amendments & Cancellations

This document describes the different activity process flows for bilateral and unilateral amendments and cancellations in MarkitWire.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire Process Flows - Amendments and Cancellations'.

MarkitWire Process Flows - Exercising Swaptions

This document describes the different activity process flows for one and two step physical swaption exercises in MarkitWire.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire Process Flows - Exercising Swaptions'.

MarkitWire Equity Swap - Queuing of Position Change Specification

This document provides a clear description of how MarkitWire handles the queuing and processing of Position Changes in an affirmation flow as well as a matching flow.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'Equity Swap - Queuing of Position Change Specification'.

MarkitWire User Guides & Cookbooks

All MarkitWire User Guides & Cookbooks can be found on the MarkitSERV documentation portal https://products.markit.com.

The following MarkitWire user guides and cookbooks are also available:

MarkitWire Trader User Guide

MarkitWire Trader allows deals to be created by dealers and legally confirmed by dealers or end users as part of the deal process. This document provides a detailed guide for using MarkitWire Trader.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire Trader User Guide'.

MarkitWire Tracker User Guide

MarkitWire Tracker is a monitoring tool that enables users to track and view deals within the MarkitWire system. This document provides a detailed guide for using MarkitWire Tracker.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire Tracker User Guide'.

MarkitWire API: Dealer Cookbook

The MarkitWire Dealer API allows deals to be created by dealers and legally confirmed by dealers or end users as part of the deal process. This document provides a detailed guide for using the MarkitWire API as a dealer.

It is available on MarkitSERV documentation portal https://products.markit.com by searching for 'MarkitWire API Dealer Cookbook'.

MarkitWire API: Broker Cookbook

The MarkitWire Broker API allows deals to be submitted by brokers, then legally confirmed by dealers or end users. This document provides a detailed guide for using the MarkitWire API as a broker.

Appendix B: Contacting Markit Support

The following has been provided in case additional information is required.

Markit Client Services for Production support:

MarkitSERV Client Services can be contacted at the following regional phone numbers:

UK/Europe:	+44 (0)84 4994 7378 / +44 203 367 0554
North America:	+1 877 765 8737
Asia Pacific:	+65 6922 4302
Japan:	+81 (0)3 6262 1885

Alternatively, you can email MarkitSERV Client Services at the following address:

support@markitserv.com

Markit Client Services for UAT / Implementation support:

Markit Client Services for UAT / Implementation can be contacted at the following address:

mserv_uat@markitserv.com