MSC Message

Attn of:	☑ Eurex Repo IT Administrators ☑ Eurex Repo Business Coordinators	Date: 06.12.2007	Pages: 7
Market:	☑ Swiss Triparty Repo Market ☑ Euro Repo Market	☑ Swiss Primary Bond Market ☑ Eurex SecLend Market	
Authorised by:	Arnold Josef		High Priority
Subject:	**Advance Notice of Changes to the PAPI and to the OTX Specification for Eurex Repo Release 9.1.x**		

G:\gbi\msc\msc mgmt\GRP_FAX\Fx7\SWX-MSC-FAX-20071206E-02.doc

Preliminary Remark: This message is only relevant to participants using the Participant Application Programming Interface (PAPI) or the Own Trade Extractor (OTX).

Dear Participants

The next Eurex Repo Release 9.1.x, which is **mandatory**, will introduce some changes to the **Participant Application Programming Interface (PAPI)** and the **Own Trade Extractor (OTX)**.

According to the tentative implementation schedule, the Eurex Repo Release 9.1.x will go in 'Membertest' on 7 January 2008 and in 'Production' on 10 March 2008. We will, of course, inform you of the final schedule and all the migration details in due course.

Please make sure that your application developers are informed of the changes to the PAPI and to the OTX, so that they can make any necessary changes to your applications in time for the start of the Membertest phase.

Changes to the Own Trade Extractor (OTX)

The following changes have been made to the OTX interface description:

Changes	Content
Tags 26, 39	Changed description relevant to Open, Variable and Open-Variable repos.
Tags 92, 93, 94, 95	Added new fields relevant to to Open, Variable and Open-Variable repos.

Note that the complete new version of the 'Own Trade File Interface Specification' (Doc Ref: ERX-XR-REP-AFI-207/E) is attached to this message.

Changes to the Participant Application Programming Interface (PAPI)

The **new classes and methods** introduced in release 9.1.x are described on the following pages. The **complete 9.1.x PAPI description** will be made available for download from the secure 'Membertest' download website with the start of the 'Membertest' phase.

Important Notes:

- **All PAPI users will have to adapt and recompile their Participant Own Applications (POAs).**

If you have any questions, please contact your Eurex Repo Technical Helpdesk:

Geneva:	Tel: +41 (0)58 854 20 28
London:	Tel: +44 (0)20 7864 4334
Zurich:	Tel: +41 (0)58 854 24 88
mailto:	techhelp@eurexrepo.com

Regards,
Eurex Repo Technical Helpdesk

PAPI Changes Introduced with Eurex Repo Release 9.1.x

swx.ric.papi.Attributes

Added constants

```
/** Repo.ContractBodyDetails: repo rate index offset*/
public static final String REPO_RATE_INDEX_OFFSET =
"REPO_RATE_INDEX_OFFSET";

/** Repo.ContractBodyDetails: index */
public static final String INDEX = "INDEX";

/** The pending repo rate, i.e. post trade CCP rate modification that is
not yet active. */
public static final String PENDING_REPO_RATE = "PENDING_REPO_RATE";

/** The date of the last rate fixing. */
public static final String REFERENCE_RATE_DATE = "REFERENCE_RATE_DATE";

/** The repo rate of the last rate fixing. */
public static final String REFERENCE_RATE = "REFERENCE_RATE";
```

Removed constants

```
/** Repo: price index */
public static final String REPO_PRICE_INDEX = "REPO_PRICE_INDEX";

/** Repo: price index family */
public static final String REPO_PRICE_INDEX_FAMILY =
"REPO_PRICE_INDEX_FAMILY";
```

swx.ric.papi.ContractType

Added methods

```
/**
* Returns all indices of the passed indexList that are known by this
* contract type.
* If indexList == null then all known indices of the contract type are
* returned.
 * If the indexList is provided but empty, then an empty list will be
 * returned.
 */
public List<Index> getIndices(List<Index> indexList)

/**
 * Returns "true" if the passed index is contained by this
 * contract type. If the passed index is "null" or if it is
 * not contained by this contract type, then "false" is
 * returned.
 */
public boolean containsIndex(Index index)

/**
 * The repo rate must be a multiple of this value.
 */
public Number getRepoRateMultipleOf()
```

swx.ric.papi.ErmDate

Added method

```
/**
 * Utility method for displaying dates in the current locale but with
 * non default formats.
 *
 * @param  date The date one wishes to represent as a string.
 * @param  aDateFormat The custom date format.
 * @return The current locale representation of the Date object   passed
 *         as a parameter.
 */
public static String asLocaleDate(Date date, DateFormat aDateFormat)
```

swx.ric.papi.ErmEnum

Added enum values to SettlementFunctionCode

```
MODIFY_RATE                ( 6, "Modify Rate", "RA"),
SET_REPURCHASE_DATE        ( 7, "Set Repurchase Date", "SR"),
INQUIRE_TRADE              ( 8, "Inquire Trade", "IT"),
REPLAY                     ( 9, "Replay", "RP");
```

swx.ric.papi.Index

Added method

```
/**
 * Compares the code of the two passed indices i1 and i2 and returns the
 * result of string compare with ingore case.
 */
public static int compare(Index i1, Index i2)
```

swx.ric.papi.IndexAdmin

Added method

```
/**
 * Returns all {@link Index} available. The list will be sorted by the
 * code of the indices.
 */
public List<Index> getAll()
```

swx.ric.papi.ValidationCodeBundle

Added validation codes

```
{ "1467", "close repo or modify rate" },
{ "1468", "rate not fixed" },
{ "1469", "missing organisation closing" },
{ "1470", "closing not equal increment" },
{ "1471", "closing past date increment" },
{ "1472", "closing past cutoff time" },
{ "1473", "repo already closed" },
{ "1474", "Mastertrade cannot be modified" },
{ "1822", "Rate fixing first date must not be specified" },
{ "1823", "Rate fixing number of periods must be one for repo rate type
           floating" },
{ "1824", "Rate fixing number of periods must be null for repo rate type
           negotiable for IOI" },
{ "1825", "Rate fixing period code must be negotiable for repo rate type
           negotiable for IOI" },
{ "1826", "Repo rate type of a quote must not be negotiable" },
```

swx.ric.papi.repo.ContractBodyDetails

Added method

```
/**
 * Returns the index referenced by reference index code.
 */
public Index getIndex()
```

swx.ric.papi.repo.OwnTradeDetails

Added methods

```
/**
 * Returns the reference index rate which is calculated as
 * follows:
 *       referenceRate = repoRate – repoRateIndexOffset
 * If the subtrahend or the minuend is not available, then null will
 * be returned.
 */
public Decimal.RepoRate getReferenceRate()

/**
 * Returns the pending repo rate, i.e. the rate modification that will
 * become the next business day.
 */
public Decimal.RepoRate getPendingRepoRate()

/**
 * The date that the value for the Reference Index was last fixed.
 */
public ErmDate getReferenceRateDate()

/**
 * Returns the reason of why the {@link SettlementStatusCode} and the
```

```
  * {@link SettlementTradeCode} has changed.
  */
 public String getSettlementReasonCode()

 /**
  * Returns "true", if the trade has been updated
  * unilaterally. Currently this is only the case when an
  * open trade is closed.
  */
 public boolean isUnilateralTradeClosure()

 /**
  * Returns "true" if this trade is a {@link
  * ErmEnum.RepoTradeType#MASTER_TRADE}, else
  * "false" will be returned.
  */
 public boolean isMasterTrade()

 /**
  * Returns "true" if repo rate modification is allowed
  * judging by the trade details' values.
  */
 public boolean modifyRepoRateAllowed()

 /**
  * Returns "true" if this trade is open and my be closed
  * according to the trade details' values.
  */
 public boolean closeOpenTradeAllowed()
```

swx.ric.papi.repo.PublicTradeDetails

(Parent class of OwnTradeDetails)

Added method

```
 /**
  * Unlike in pre-trading, where RepoRate either contains the repo rate
  * index offset (in Basepoints BP) or the repo rate (in %) depending on
  * the repoRateType, do Own Repo Trades contain a dedicated attribute
  * for the repo rate index offset. Thus we have to overwrite the
  * {@link ContractBodyDetails#getRepoRateIndexOffset()} and the
  * {@link ContractBodyDetails#getRepoRate()} methods of
  * {@link ContractBodyDetails}.
  */
 public Decimal.RepoRate getRepoRateIndexOffset()
```

swx.ric.papi.repo.TradeModifyRequestDetails

Added method

```
 /**
  * Returns the repo rate if the request is due to a repo rate
  * modification.
  */
 public Decimal.RepoRate getRepoRate()
```

swx.ric.papi.repo.TradeModifyRequestEntryDetails

Added methods

```
/**
 * Returns the repo rate set on the request.
 */
public Decimal.RepoRate getRepoRate()

/**
 * Sets the repo rate on the request.
 */
public void setRepoRate(Decimal.RepoRate aRepoRate)
```

swx.ric.papi.util.PapiUtils

Added methods

```
/**
 * @param repoRateType
 * @return whether the given repoRateType is of a "relative" type, i.e.
 *         not FIXED
 */
public static boolean isRepoRateTypeRelative(RepoRateType repoRateType)

/**
 * @return true if given custodyType is GCPooling
 */
public static boolean isCCPPooling(CustodyType custodyType)

/**
 * @return whether cashAmount is required or not (securitySize).
 *         It is active if xetraCashAmount is enabled (setting in
 *         the config file) or the current custodyType supports
 *         'CCP Pooling'.
 */
public static boolean isCashAmountActive(Market market,
    CustodyType custodyType)
```