

CITY OF WINNIPEG

The Timekeeping Interface

User's Reference Guide

David Sharman

1/27/2014

Table of Contents

The Purpose of the Timekeeping Interface (TKI)	1
Service Operations Provided.....	1
CNX_TKI_PROCESS_SCHD_ACTUALS	1
CNX_TKI_PROCESS_TIME_ENTERED	2
CNX_TKI_GET_TK_STATE	2
CNX_TKI_GET_EMPLID.....	2
Web Service Security	2
Node Level	2
User Level.....	3
SSL Considerations	3
The Request Structures.....	3
Common Elements.....	3
HTTP Request Header	3
SOAP.....	4
The Process Schedules and Actuals Request Structure	5
The Employee Level	5
The Date Under Report Level.....	6
The Schedules / Actuals Level.....	6
The Required Information Answer Level	8
The Process Time Entered Request Structure	9
The Employee Level	9
The Date Under Report Level.....	9
The Time Entered Level.....	9
The Required Information Answer Level	11
The Get Timekeeping State Request Structure.....	11
The Timekeeping State Level	11
The Get Employee ID Request Structure	12
The Operator ID Level	12
The Response Structures	12
Common Elements.....	12

HTTP Response Header	12
SOAP.....	12
The Process Schedules and Actuals Response Structure	13
The Employee Level	14
The Date Under Report Level.....	15
The Schedule / Actuals Level	15
The Time Entered Level.....	16
The Required Info Answers Level.....	17
The Time Reported Level	17
The Time Reported Messages Level.....	19
The Required Information Level	20
The Required Info Question Level.....	20
The Required Info Question Possible Values Level.....	21
The Time Entered Response Structure	21
The Get Timekeeping State Response Structure	22
The Employee Level	22
The Plan Balance as of Now Level.....	23
The Date Under Report Level.....	23
The Job Level.....	24
The Name Level.....	24
The Time Reporter Level.....	24
The Schedule Level.....	25
The Plan Balance as of a DUR Level	26
The Time Entered Level.....	27
The Time Reported Level	28
The Time Reported Messages Level.....	29
The Reported Time Level	29
The Get Employee ID Response Structure	30
The Employee Level	30
The Response Structure and the User Specified Response Level.....	31
Replacing Data for the DUR	32

Generic Time Codes, Subcodes and Required Info Ids	32
ANP - Away Without Pay.....	32
AWP - Away With Pay	33
BTC - Banked Time Cash Out.....	33
BTT - Bank Time Off	33
CPW - Compressed Work Hrs To Bank.....	33
CTM - Card Time	34
DIS - Disinterment.....	34
EDU - Education and Training	34
FLD - Flood	35
GRR - Gradual Returns	35
LEV - Leaves.....	35
REG - Regular	36
ROT - Overtime	36
SIC - Sick	37
SNT - Sent Home - Weather/Breakdown	38
SOT - Special Overtime.....	38
STB - Standby	39
STH - Statutory	39
TSB - Transit Standby	40
VAC - Vacation.....	40
VWL - Vacation/WAPSO Days/Long Serv.	41
WST - Work Stoppage	41

The Purpose of the Timekeeping Interface (TKI)

The Timekeeping Self Service application (TKSS), implemented in the spring of 2012, allows CUPE and WAPSO employees to report Timekeeping related information such as time worked, overtime, vacations, absences due to illness, etc. TKSS was designed to allow the employee to “report the facts” and the interpretation of these facts is done by TKSS based on the CUPE and WAPSO collective bargaining rules agreements. The interpretation of these rules creates “pre-reported” time which is then sent to PeopleSoft’s Time and Labour application where it is turned into Reported Time.

There are a number of scheduling / time tracking applications used by various departments around the City including the InIn application at 311 and TKMMS at Public Works. Rather than have employees whose time is tracked through these applications enter this same time into TKSS, it would be more efficient to have the data from these systems sent to the Timekeeping rules that are inside TKSS. As well, we want to ensure that the interpretation of the CUPE and WAPSO collective bargaining rules is done in only a single place (ie. in TKSS).

The Timekeeping Interface (TKI) allows external applications to take advantage of the Timekeeping rules that have been implemented in TKSS. These external applications will send transactions to the TKI and the TKI will use the Timekeeping rules in TKSS to create the appropriate pre-reported time. In effect, the TKI will mimic what would happen if the employee had reported the information into TKSS.

Service Operations Provided

A single service, CNX_TKI, has been created for the Timekeeping Interface.

See the production version of the Web Services Description Language for this service at http://cowsvwb17?.ad.cityofwpg.org:????/PSIGW/PeopleSoftServiceListeningConnector/CNX_TKI.1.wsd!

The service CNX_TKI contains the four service operations that make up the Timekeeping Interface.

- CNX_TKI_PROCESS_SCHD_ACTUALS
 - Process the Schedule / Actuals transaction passed in
- CNX_TKI_PROCESS_TIME_ENTERED
 - Process the Time Entered transaction passed in
- CNX_TKI_GET_TK_STATE
 - Get the Timekeeping State for the employee ID and date range passed in
- CNX_TKI_GET_EMPLID
 - Find the PeopleSoft Employee ID for the Active Directory ID passed in

CNX_TKI_PROCESS_SCHD_ACTUALS

This is the primary service operation that will be used by the external applications. When an external application wants to report some time for an employee, it sends a Process Schedules and Actuals

transaction to PeopleSoft HR. PeopleSoft HR transforms the Schedule and Actuals transaction to a Time Entered transaction and processes the Time Entered transaction (see the next service operation). If the processing of the transaction is successful, the service creates the appropriate “pre-reported” time, updates the PeopleSoft HR database and returns some information to the external application.

CNX_TKI_PROCESS_TIME_ENTERED

This is a service operation that the external applications will not likely be calling, but has been provided in case there are rare occasions or possibly future external applications that will want to use the Timekeeping Interface using this service operation. If this service is used, the external application sends a Process Time Entered transaction to PeopleSoft HR. PeopleSoft HR processes the Time Entered transaction and if it is successful, creates the appropriate “pre-reported” time, updates the PeopleSoft HR database and returns some information to the external application.

CNX_TKI_GET_TK_STATE

This service operation allows an external application to get information about an employee related to Timekeeping. At this point, no external application has yet expressed a need to do this, but the service has been provided in case an external application wants to present an employee’s state on one of its pages or if the external application wants to do some validation (eg. check an employee’s vacation balance before sending a vacation transaction through the TKI).

CNX_TKI_GET_EMPLID

This service operation has been provided to allow the translation of an Active Directory (AD) ID to a PeopleSoft Employee ID. The employees in the 311 application are uniquely identified by their AD ID. The above service operations require that an employee be uniquely identified by the PeopleSoft HR EMPLID. This service operation allows the translation of an AD ID to a PeopleSoft Emplid.

Web Service Security

All of the above four service operations are subject to two levels of security

- Node level
- User level

Node Level

The external application invoking the service operation is identified as an Integration Broker node in PeopleSoft HR. When an external application invokes the service operation, it must provide the name of the node and the password associated with the node. The node name and password assigned in PeopleSoft HR will be given to the developers of the external applications.

See the Request Structure section below for details of how this information is sent as part of the service operation invocation.

User Level

In addition to providing node level credentials, the external application must also provide credentials for the user that is making the request. That is, it must supply the PeopleSoft Operator ID and PeopleSoft password for that Operator Id. The PeopleSoft Operator ID must be configured within PeopleSoft HR to be allowed to use the service operation being called and the PeopleSoft Operator ID must be configured to be allowed to report time for the employee for whom the transaction is being submitted.

Yet to be determined is if this Operator ID will be a generic PeopleSoft ID created for each of the external applications or if it will be an existing PeopleSoft operator IDs that are for employees who are the supervisors of the employees being processed.

See the Request Structure section below for details of how this information is sent as part of the service operation invocation.

SSL Considerations

Note that the invocation of the service operations is currently not using HTTPS. That is, the requests are being sent over the intranet in a plain text format rather than encrypted. This may be a concern for some (including me). If we cannot invoke these service operations using HTTPS in the short term, we may want to lean towards using newly created generic IDs rather than existing employee operator IDs. Note that up until 2012, when employees logged in to PeopleSoft their IDs and passwords were NOT encrypted as they went over the intranet.

The Request Structures

Common Elements

While each of the four service operations has its own request structure, there are a number common elements shared across all of the service operations.

HTTP Request Header

The HTTP Header contains the Node Level credentials (see above). These node level credentials are contained in the HTTP Header called SOAPAction. The SOAPAction contains

- The name of the service operation
- The name of the PeopleSoft node invoking the service operation
- The password for this PeopleSoft node

For example, to invoke the Process Schedules and Actuals service operation, the HTTP Header would contain:

SOAPAction: #CNX_TKI_PROCESS_SCHD_ACTUALS.v1#CNX_311#DaveTest

where ...

- *CNX_TKI_PROCESS_SCHD_ACTUALS.v1* is the name and version of the service operation to invoke
- *CNX_311* is the name of the 311 PeopleSoft node
- *DaveTest* is the password for the PeopleSoft node

A complete HTTP Header, most of which should be generated by the tool used in the external application based on the TKI Service's WSDL:

```
POST /PSIGW/PeopleSoftServiceListeningConnector HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: #CNX_TKI_PROCESS_SCHD_ACTUALS.v1#CNX_311#DaveTest
Content-Length: 1952
Host: crowsvdbw171.ad.cityofwpg.org:5224
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

SOAP

All service operations request structures are in XML and are SOAP-based (Simple Object Access Protocol). That is, all requests contain:

- A SOAP envelope
- A SOAP Header
- A SOAP Body

SOAP Envelope

The SOAP envelope is a container for the SOAP message. It contains the Namespace definitions that will be used in the XML request.

SOAP Header

The SOAP Header contains the User Level credentials (see above). That is, the PeopleSoft Operator ID and the PeopleSoft password associated with this Operator ID.

Sample SOAP Header with the user credentials bolded:

```
<soapenv:Header>
  <wsse:Security soap:mustUnderstand="1" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-1" xmlns:wsu="http://docs.oasis-
  open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>GJANSZ</wsse:Username>
      <wsse:Password>TestIB09</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
```


SOAP Body

The SOAP Body contains the actual content of the request. See below for the details of request structure for each of the four service operations.

The Process Schedules and Actuals Request Structure

The Process Schedules and Actuals request structure is based on the PeopleSoft Message Container TKScheduleActualsRequest (CNX_TKI_SCHD_ACTL_RQST_MC.V1) which is based on the PeopleSoft Message Part TKScheduleActualsRequest_MP (CNX_TKI_SCHD_ACTL_RQST_MP).

The request structure contains the following hierarchy:

- The Employee Level: TKTEmployee (1)
 - The Date Under Report Level: TKDateUnderReport (1,M)
 - The Schedule / Actuals Level: TKTScheduleAndActuals (0,1,M)
 - The Required Info Answers Level: TKTReqInfoAnswers (0,1,M)

The Employee Level

TKTEmployee (CNX_TKT_EMPL)

There must be exactly 1 Employee Level entry in each transaction.

- EMPLID
 - The PeopleSoft employee ID that uniquely identifies this employee in PeopleSoft
 - 6 digits
 - Required
- EMPL_RCD
 - The employee record number that allows for the employee to have multiple relationships with the City of Winnipeg
 - As we do not currently have “multiple jobs” in PeopleSoft, this will always be 0
 - Required
- CNX_TK_CRT_RPTD_TM
 - A Yes / No flag indicating if the transaction should create reported time (ie. update the PeopleSoft database) if the transaction is processed successfully
 - Y indicates that the PeopleSoft database should be updated
 - N indicates that the PeopleSoft database should not be updated (ie. the transaction is being processed in as “audit mode” or a “report only mode”).
 - Optional, Default: Y
- CNX_TK_RESP_LEVEL
 - A flag indicating the level of detail that is to be provided in the response to processing the transaction
 - L – Low –just a success flag is returned in response to successfully processing the transaction. If the transaction fails, more detail is provided to allow the caller to understand why the transaction could not process successfully.

- M – Medium – a success flag and the time entered, time reported and any messages created by the TKI are returned in response to successfully processing the transaction. If the transaction fails, the Required Info or rejection reason is returned
- H – High – like medium, plus the details of the transaction request are echoed back in the response
- Optional, Default: M
- DEBUG
 - A Yes / No flag to turn debugging on
 - Y indicates that debugging should be turned on. This results in some Timekeeping State information being written to files to facilitate debugging
 - N indicates that debugging is turned off
 - Optional, Default: N

The Date Under Report Level

TKTDateUnderReport (CNX_TKT_DUR)

Under the Employee Level, there must be 1 or many Dates Under Report entries.

- DUR
 - The date for which the time is being reported
 - Format: yyyy-mm-dd
 - Required

The Schedules / Actuals Level

TKTScheduleAndActuals (CNX_TKT_SCH_ACT)

Under each DUR, there can be 0, 1 or many Schedule / Actuals entries.

- CNX_SCH_ACT_IND
 - A flag indicating if this line represents the time that an employee was scheduled to work or when the employee actually worked
 - S – Scheduled
 - A – Actual
 - Required
- CORREL_ID
 - An correlation identifier that will relate a specific request entry to a specific line or lines in the response
 - The correl ID should not be specified for a Scheduled line; it will be ignored
 - The correl ID should be specified for an Actuals line
- START_DTTM
 - The start date and time of the schedule or actual time being reported
 - Format: yyyy-mm-dd hh:mi:ss where hh is a 24 hour clock; all values must be padded on left with zeroes

- Required
 - END_DTTM
 - The end date and time of the schedule or actual time being reported
 - Format: yyyy-mm-dd hh:mi:ss where hh is a 24 hour clock; all values must be padded on left with zeroes
 - Required
 - CNX_TK_TC_CD
 - The generic time code for the schedule or actual line being reported
 - Examples include REG (Regular time), SIC (Sick time), VAC (Vacation time), Leaves (LEV), etc.
 - For a Scheduled line, this should always be REG
 - For an Actuals line, see below for a list of generic time codes
 - Required
 - CNX_TK_SC_CD
 - The generic time sub code for the actual line being reported
 - Not all generic time codes require a subcode
 - For example, there is no subcode for REG, SIC, etc, but for Leaves (LEV), an example subcode is BLV (Bereavement Leave).
 - For a Scheduled line, this should always be blank
 - For an Actuals line, see later in this document for a list of generic time sub codes and the generic time codes to which they are applicable
 - Optional, dependent on type of line (schedule or actuals) and generic time code
 - CNX_TK_STC_GRP
 - The special Timekeeping Group indicates that the time for this transaction should be processed using special rules for that group. For example, if someone is a member of the group SYNERGEN, then additional information such as the Synergen Work Order number must be provided
 - This is unlikely going to be used for 311 or TKMMS
 - Optional
 - JOBCODE
 - If an employee is acting in another classification for the work being reported, the job code of the classification the employee is acting in must be reported
 - This is only applicable for Actuals entries
 - Optional
 - ACCT_CD
 - If the time being reported is supposed to be charged to an account code (combination of chart fields including department ID, account number, fund code, etc) then an account code should be specified
 - This is only applicable for Actuals entries
 - Optional
 - CNX_TK_WCB_NBR
-

- If the time being reported is related to a WCB claim, then the Workers Compensation Board number should be specified
- This is only applicable for Actuals entries
- Optional
- CNX_TK_STC_FLD1
 - If the time being reported is supposed to be attributed to some work order or task, then the Special Time Code fields will be filled in.
 - Special Time Code Field 1 holds the Maximo Work Order Number
 - This is only applicable for Actuals entries
 - Optional
- CNX_TK_STC_FLD2
 - If the time being reported is supposed to be attributed to some work order or task, then the Special Time Code fields will be filled in.
 - Special Time Code Field 2 holds the Maximo Work Location
 - This is only applicable for Actuals entries
 - Optional
- CNX_TK_STC_FLD3
 - If the time being reported is supposed to be attributed to some work order or task, then the Special Time Code fields will be filled in.
 - Special Time Code Field 3 holds the Synergen Work order????
 - This is only applicable for Actuals entries
 - Optional
- CNX_TK_COMMENT
 - A comment for the entry
 - This is only applicable for Actuals entries
 - Optional

The Required Information Answer Level

TKTReqInfoAnswers (CNX_TKT_RQ_ANSW)

Under each Actuals entry, there can be 0, 1 or many Required Info Answer entries. A Required Info answer entry provides the answer to a Required Information question. For example, to process an Overtime entry, the system needs to know if the employee wants to bank his or her overtime. The request can provide the answer Y for the Required Info ID isBankOt. For a complete list of Required Info Identifiers, see later in this document.

- CNX_TK_REQ_INFO_ID
 - The identifier for the specific Required Info Id for which the answer is being provided
 - Required
- CNX_TK_ANSWER_TYPE
 - What type of answer is being provided

- Note that the answer type is returned to the caller if not all Required Info has been provided (and cannot be derived)
 - Valid choices are:
 - YesNo
 - Number
 - Text
 - Date
 - Time
 - DateTime
 - FromList
 - FromListNbr
 - Required
- CNX_TK_ANSWER
 - The answer to the Required Info question
 - Based on the answer type, the answer must be in the correct format
 - Number: digits 0 to 9
 - Date: yyyy-mm-dd
 - Time: hh:mi:ss where hh is a 24 hour clock
 - DateTime yyyy-mm-dd hh:mi:ss where hh is a 24 hour clock
 - For answer type FromList or FromListNbr, the answer must be one provided in the Possible Values list
 - Required

The Process Time Entered Request Structure

The Process Time Entered request structure is based on the PeopleSoft Message Container TKTimeEnteredRequest (CNX_TKI_TIME_ENTD_RQST_MC.V1) which is based on the PeopleSoft Message Part TKTimeEnteredRequest_MP (CNX_TKI_TIME_ENTD_RQST_MP.V1).

The request structure contains the following hierarchy:

- The Employee Level: TKTEmployee (1)
 - The Date Under Report Level: TKTDateUnderReport (1,M)
 - The Time Entered Level: TKTimeEntered (0,1,M)
 - The Required Info Answers Level: TKTReqInfoAnswers (0,1,M)

The Employee Level

See the Employee Level for *The Process Schedules and Actuals Request Structure* above.

The Date Under Report Level

See the Date Under Report Level for *The Process Schedules and Actuals Request Structure* above.

The Time Entered Level

TKTimeEntered (CNX_TKT_TM_ENTD)

Under each DUR, there can be 0, 1 or many Time Entered entries. Each Time Entered entry is the equivalent to a single line entered in the Timekeeping Self Service application.

Most fields are the same as the fields for the Schedule and Actuals level described above in *The Process Schedules and Actuals Request Structure*. I will list all of the fields here but only put in additional notes if they are different from the notes above.

- CORREL_ID
- CNX_TK_TC_CD
- CNX_TK_SC_CD
- TL_QUANTITY
 - The number of hours being reported
- CNX_TK_SHIFT_BTN
 - A Yes / No indicator as to whether the time being reported should be considered for shift premium
 - This is only applicable if the Generic Time Code (CNX_TK_TC_CD) is REG (Regular Pay)
 - Y – consider the time for shift premium
 - N – do not consider the time for shift premium
 - Optional, default is N
- CNX_TK_PEER_GRP
 - A Yes / No indicator as to whether the time being reported should be considered for peer group (not sure what this means – will follow-up with DLR???)
 - This is only applicable if the Generic Time Code (CNX_TK_TC_CD) is REG (Regular Pay)????
 - Y – consider the time for peer group pay
 - N – do not consider the time for peer group pay
 - Optional, default is N
- CNX_TK_STC_GRP
- JOBCODE
- ACCT_CD
- CNX_TK_WCB_NBR
- CNX_TK_STC_FLD1
- CNX_TK_STC_FLD2
- CNX_TK_STC_FLD3
- CNX_TK_START_TIME
- CNX_TK_END_TIME
- CNX_TK_START_DTTM
- CNX_TK_END_DTTM
- CNX_TK_COMMENT

The Required Information Answer Level

See the Required Information Answer Level for *The Process Schedules and Actuals Request Structure* above.

The Get Timekeeping State Request Structure

The Get Timekeeping State request structure is based on the PeopleSoft Message Container TKCurrentStateRequest (CNX_TKI_CURRENT_STATE_RQST_MC.V1) which is based on the PeopleSoft Message Part TKCurrentStateRequest_MP (CNX_TKI_CURRENT_STATE_RQST_MP.V1).

The request structure contains the following one-level hierarchy:

- The Timekeeping State Level: CNX_TKT_TKST (1)

The Timekeeping State Level

TKTimekeepingState (CNX_TKT_TKST)

There must be exactly one The Timekeeping State Level entry in each transaction.

- EMPLID
 - The PeopleSoft employee ID that uniquely identifies this employee in PeopleSoft
 - 6 digits
 - Required
- EMPL_RCD
 - The employee record number that allows for the employee to have multiple relationships with the City of Winnipeg
 - As we do not currently have “multiple jobs” in PeopleSoft, this will always be 0
 - Required
- FROM_DATE
 - The earliest date for which the Timekeeping State is to get information
 - Format: yyyy-mm-dd
 - Required
- TO_DATE
 - The latest date for which the Timekeeping State is to get information
 - Format: yyyy-mm-dd
 - The From Date must be less than or equal to the To Date
 - The From Date and To Date must be within 14 days of each other
 - Required
- DEBUG
 - A Yes / No flag to turn debugging on
 - Y indicates that debugging should be turned on. This results in some Timekeeping State information being written to files to facilitate debugging
 - N indicates that debugging is not turned on
 - Optional, Default: N

The Get Employee ID Request Structure

The Get Timekeeping State request structure is based on the PeopleSoft Message Container TKGetEmplidRequest_MC (CNX_TKI_GET_EMPLID_RQST_MC.V1) which is based on the PeopleSoft Message Part TKGetEmplidRequest_MP (CNX_TKI_GET_EMPLID_RQST_MP.V1).

The request structure contains the following one-level hierarchy:

- The Operator Id Level: CNX_TKT_OPRID (1)

The Operator ID Level

TKTOperatorID (CNX_TKT_OPRID)

There must be exactly one Operator ID Level entry in each transaction.

- OPRID
 - The operator ID (aka the Active Directory ID) of the employee for whom we want to find the employee ID
 - Required

The Response Structures

Common Elements

While each of the four service operations has its own response structure, there are a number common elements shared across all of the service operations.

HTTP Response Header

The HTTP Header contains, among other things, the Response Status. A Status of 200 indicates that the service completed successfully, at least from an HTTP perspective. The success or failure of the transaction from a TKI perspective will be discussed shortly. A status of anything other than 200 indicates that the transaction failed from an HTTP perspective.

SOAP

All service operations response structures are in XML and are SOAP-based (Simple Object Access Protocol). That is, all requests contain:

- A SOAP envelope
- A SOAP Body

SOAP Envelope

The SOAP envelope is a container for the SOAP message. It contains the Namespace definitions that will be used in the XML request.

SOAP Body

The SOAP Body contains the actual content of the response. See below for the details of response structure for each of the four service operations when the transaction is successful (from an HTTP perspective).

Fault Response Structure

If the transaction fails from an HTTP perspective, then a Fault Response is generated. The Fault Response structure contains the following elements within the response SOAP Body:

```
<SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring></faultstring>
  <detail>
    <IBResponse type="error">
      <DefaultTitle>Integration Broker Response</DefaultTitle>
      <StatusCode>nn</StatusCode>
      <MessageID>nnn</MessageID>
      <DefaultMessage>xxx</DefaultMessage>
      <MessageParameters>
        <Parameter/>
      </MessageParameters>
    </IBResponse>
  </detail>
</SOAP-ENV:Fault>
```

The Default Message contains the specific error and, if applicable, the call stack.

The Process Schedules and Actuals Response Structure

The Process Schedules and Actuals response structure is based on the PeopleSoft Message Container TKScheduleActualsResponse (CNX_TKI_SCHD_ACTL_RESP_MC.V1) which is based on the PeopleSoft Message Part TKScheduleActualsResponse_MP (CNX_TKI_SCHD_ACTL_RESP_MP.V1).

The response structure contains the following hierarchy:

- The Employee Level: TKSEmployeeResp (1)
 - The Date Under Report Level: TKSDateUnderReportResp (1,M)
 - The Schedule / Actuals Level: TKTScheduleAndActualsResp (0,1,M)
 - The Time Entered Level: TKTimeEnteredResp (0,1,M)
 - The Required Info Answers Level: TKTRRequiredInfoAnswerResp (0,1,M)
 - The Time Reported Level: TKRTTimeReportedResp (0,1,M)
 - The Time Reported Messages Level: TKRTTimeReportedMessagesResp (0,1,M)
 - The Required Info Level: TKRRRequiredInfoResp (0,1)
 - The Required Info Question Level: TKRRRequiredInfoQuestionResp (1,M)

- The Required Info Answer Possible Values:
TKRRequireInfoQuestionPssblVal (0,1,M)

The **yellow** highlighted lines contain an “echo” of the request. The **green** highlighted lines contain the response from processing the request.

Note that if a level has no data in it, then it will not be part of the response structure.

Note also that the Time Reported and Time Reported Messages levels are mutually exclusive with the Required Info Level. If a Time Entered transaction (which resulted from the transformation of the Schedule / Actuals transaction) is successfully processed, then the Time Reported and Time Reported Messages levels will be filled in and the Required Info Level will be empty. If a Time Entered transaction is NOT successfully processed, then the Time Reported and Time Reported Messages levels will be empty and the Required Info level will (typically) be populated indicating what information was missing that prevented the transaction from being processed successfully.

And finally, note that the level of detail of the response is controlled by the values specified in the CNX_TK_RESP_LEVEL field in the request. The above structure is the complete structure. It will be returned if the requester specifies H (High) for the CNX_TK_RESP_LEVEL. If the requester specifies M (Medium), then the Schedule / Actuals level will be excluded from the response, and the Time Entered level will contain only a subset of the possible fields. If the requester specifies L (Low), then only the CNX_SUCCESS flags will be returned. The level at which the success flags are returned depend on the overall success of the transaction and the level at which the database is being updated. See later in the document for more details.

The Employee Level

TKSEmployeeResp (CNX_TKS_EMPL)

There is exactly one Employee Level entry in each response.

- EMPLID
 - The PeopleSoft employee ID that uniquely identifies this employee in PeopleSoft
 - 6 digits
- EMPL_RCD
 - The employee record number that allows for the employee to have multiple relationships with the City of Winnipeg
 - As we do not currently have “multiple jobs” in PeopleSoft, this will always be 0
- FROM_DATE
 - The earliest DUR of the transaction being processed
 - Format: yyyy-mm-dd
- TO_DATE
 - The latest DUR of the transaction being processed

- Format: yyyy-mm-dd
- DTTM_CREATED
 - The date and time that the transaction was processed
 - Format: yyyy-mm-dd hh:mi:ss where hh is a 24 hour clock; all values will be padded on the left with zeroes
- CNX_SUCCESS
 - A Yes / No flag indicating if the entire transaction was processed successfully from a TKI perspective
 - Y – The transaction was processed successfully
 - N – The transaction was not processed successfully
 - Information about the reason for failure is described below

The Date Under Report Level

TKTDateUnderReportResp (CNX_TKS_DUR)

There is one or many Dates Under Report under the employee level.

- DUR
 - The date for which the transaction is being processed
 - Format: yyyy-mm-dd
- CNX_SUCCESS
 - A Yes / No flag indicating if all of the transactions for a DUR were processed successfully from a TKI perspective
 - Y – All transactions for the DUR were was processed successfully
 - N – Not all transactions for the DUR were processed successfully
 - Information about the reason for failure is described below

The Schedule / Actuals Level

TKTScheduleAndActualsResp (CNX_TKT_SCH_ACT)

There are 0, 1 or many Schedule / Actuals entries under each DUR.

The schedule / actuals entries contain information about what an employee was scheduled to do and what the employee actually did.

Most fields are identical to the Schedules / Actuals Request structure. See above for more details. Most of the additional fields that are in the response but not in the request are for debugging and troubleshooting purposes and would not typically be relevant by the external application. They are described here:

- SEQ_NBR
 - The unique (within the DUR), sequential number assigned to the Schedule / Actual entry
- CNX_TK_TC_ID

- The Time Code ID is an internal ID used by the Timekeeping application. It links the employee's union code with the transaction's Generic Time Code and Subcode.
- CNX_TK_CLASS_NAME
 - The Class Name is an internal identifier of the Timekeeping Article class that is used to process the transaction
 - The TK class to use is based on the CNX_TK_TC_ID
- CNX_TKS_STATE_STTS
 - The Timekeeping State Status for this message
 - Possible values are LA – Logically Added and LD – Logically Deleted
- TCD_ID
 - The Time Collection Device identifier
 - The TCD ID is based on the external application which sent the transaction

The Time Entered Level

TKTimeEnteredResp (CNX_TKT_TM_ENTD)

There are 0, 1 or many Time Entered entries under each DUR.

The time entered entries contain information about the time that the employee worked, including the generic time code and the amount of time worked. Entries in this level are generated based on the transformation of the Schedule and Actuals lines to Time Entered lines.

Most fields are identical to the Time Entered Request structure. See above for more details. Most of the additional fields that are in the response but not in the request are for debugging and troubleshooting purposes and would not typically be relevant by the external application. Most of these are described just above and are listed here without any comments. The fields specific to the Time Entered Response structure that are not in the Time Entered Request or in the Schedule / Actuals response are described here:

- SEQ_NBR
- CNX_TK_TC_ID
- CNX_TK_CLASS_NAME
- CNX_TKS_STATE_STTS
- START_DTTM
 - The start date / time of an “actuals segment”
 - An “actuals segment” is an internal construct created by the TKI when transforming schedules and actuals lines into time entered lines
- END_DTTM
 - The end date / time of an “actuals segment”
- CNX_TK_SCH_TC_CD
 - The generic time code from the schedule line (if any) associated with an actuals segment
- CNX_SHFT_STRT_DTTM

- The earliest scheduled start date / time for a DUR
- This value always comes from a Schedule line, not from an actuals line
- CNX_SHFT_END_DTTM
 - The latest scheduled end date / time for a DUR
 - This value always comes from a Schedule line, not from an actuals line
- CNX_REG_STARTDTTM
 - The start date / time of the work for the DUR that is considered Regular (REG) work
- CNX_REG_ENDDTTM
 - The end date / time of the work for the DUR that is considered Regular (REG) work
- TCD_ID
 - The Time Collection Device Identifier for this time entered
 - The TCD ID is determined based on the external system that creates sends the transaction

The Required Info Answers Level

TKTRequiredInfoAnswerResp (CNX_TKT_RQ_ANSW)

Under the Time Entered entry, there can be 0, 1 or many Required Info Answer entries.

This level contains the user-supplied answers to Required Info questions. That is, this level is populated directly from the request while the subsequent levels are as a result of processing the request.

Most fields are identical to the Required Info Answer Request structure. See above for more details. The additional field in the response, but not the request, is for debugging and troubleshooting purposes and would not typically be relevant by the external application.

- CNX_COUNT_ANSW_USE
 - This field displays the number of times that the provided Required Info Answer was used in the processing of its parent Actuals entry
 - If the use count is 0 after the Actuals entry has been successfully processed, it means that the Required Info answer was not used in the processing of the transaction which means that it did not need to be provided. External systems may want to check this field.

The Time Reported Level

TKRTimeReportedResp (CNX_TKR_TM_RPTD)

Under the Time Entered entry, there can be 0, 1 or many Time Reported entries.

The Time Reported level is populated as a result of processing the request. This level contains the “pre-reported” time that will ultimately get converted to PeopleSoft’s Reported Time.

- ADDL_SEQ

- A unique (within a Time Entered), sequential identifier assigned by the system when the Time Reported is created
- CNX_TK_TC_GRP
 - The Time Code Group used to create this Time Reported
 - 1 – Self Service
 - 2 – Manager Entered
 - 3 – Entered by a TK Specialist
- CNX_TK_STATUS
 - The Timekeeping Status of the Time Reported
 - Time reported generated through the TKI will have a TK Status of A (Approved)
- TRC
 - The Time Reporting Code for the Time Reported
- TL_QUANTITY
 - The amount of time being reported with this Time Reported record
- JOBCODE
 - The jobcode in which the employee was acting
 - Blank indicates that there is no Acting Pay considerations
- ACCT_CD
 - The account code to which this time should be charged
- OVERRIDE_RATE
 - The rate at which this time should be paid
 - The override rate should only be specified if it is different from the employee's rate in Job
- CNX_TK_COMMENT
 - A comment associated with the Time Reported
- CNX_TK_STC_ID
 - The Timekeeping Special Time Code
 - ?????
- CNX_TK_TC_ID
 - The internal Time Code ID for the combination of union, generic time code and subcode
- USER_FIELD_1
 - User Fields allow additional information to be associated with this time reported
 - User Field 1 holds the WCB Number
- USER_FIELD_2
 - User Field 2 holds the Synergen Work Order Number for WW
- USER_FIELD_3
 - User Field 3 holds the Synergen Task Number for WW
- USER_FIELD_4
 - User Field 4 holds the Maximo Work Order Number for PPnD
- USER_FIELD_5

- User Field 5 holds the Maximo Location for PPnD
- TCD_ID
 - The Time Collection Device from which this Time Reported was created
 - As noted above, this is based on the external application which sent the transaction
- COMP_LEAV_IND
 - How the time reported affects a plan
 - CERN – Compensatory Time Off (CTO) Plan Earned
 - CTKN – CTO Taken
 - LTKN – Leave Taken
 - NO – No Plan affected
- CNX_PLAN
 - The specific compensatory time or leave plan affected
 - Eg. OVERTME, 50 (Sick), etc.
- CNX_COMP_OR_LEAV
 - A flag indicating if this time reported affects a Compensatory Time Plan (eg. overtime), a Leave Plan (eg. Vacation) or neither
 - C – Compensatory Plan
 - L – Leave Plan
 - Blank – no plan
- PLAN_DESCR
 - The description of the Plan affected

The Time Reported Messages Level

TKRTimeReportedMessagesResp (CNX_TKR_INT_TXN)

Under the Time Entered entry, there can be 0, 1 or many Time Reported Message entries.

The Time Reported Messages Level can contain both

- Messages generated by the processing of a transaction
- The question and answer dialogue that is generated by the processing of a transaction

The fields at the Time Reported Message level are:

- ADDL_SEQ
 - A unique (within a Time Entered), sequential identifier assigned by the system when the Time Reported message is created
- CNX_TKS_STATE_STTS
 - The Timekeeping State Status for this message.
 - Possible values are LA – Logically Added and LD – Logically Deleted
 - May want to delete this field too as it is not relevant to consumers
- CNX_TK_ANSWER
 - If the CNX_TK_REQ_INFO_ID is blank, this field is blank

- If the CNX_TK_REQ_INFO_ID is not blank, this field holds the answer to the question for Required Info
- CNX_TK_ANSWER_TYPE
 - If the CNX_TK_REQ_INFO_ID is blank, this field is blank
 - If the CNX_TK_REQ_INFO_ID is not blank, this field holds the type of answer
 - See subsection The Required Information Answer Level under section The Process Schedules and Actuals Request Structure above for the list of possible Answer Types
- CNX_TK_REQ_INFO_ID
 - If the entry is a message, the CNX_TK_REQ_INFO_ID is blank
 - If the entry is a request for more information, this field holds the internal Timekeeping Identifier for the Required Information
- CNX_TK_QUESTION
 - If the CNX_TK_REQ_INFO_ID is blank, this field holds a message generated by processing of the transaction
 - If the CNX_TK_REQ_INFO_ID is not blank, this field holds the verbiage of the Required Info question

The Required Information Level

TKRRequiredInfoResp (CNX_TKR_RQ_INFO)

Under the Time Entered entry, there can be 0 or 1 Required Info entries.

If, as the Timekeeping rules are processing a transaction, they determine that additional information is required to continue processing the transaction, a Required Info entry will be created in the response.

The fields at the Required Info level are:

- ADDL_SEQ
 - A unique (within a Time Entered), sequential identifier assigned by the system when the Required Info is created
- CNX_TK_REQ_INFO_ID
 - The internal Timekeeping Identifier for the Required Information

The Required Info Question Level

TKRRequiredInfoQuestionResp (CNX_TKR_RQ_QSTN)

Under a Required Info entry, there can be 1 or Many Required Info Question entries. Note that while Many should be allowed for, in the current implementation of Timekeeping, there will only be 1 question ID.

The fields at the Required Info Question level are:

- CNX_TK_QUESTION_ID
 - The internal identifier for the question

- CNX_TK_QUESTION
 - The specific question that needs to be answered
- CNX_TK_ANSWER_TYPE
 - The type of answer that needs to be provided for this question
 - See subsection The Required Information Answer Level under section The Process Schedules and Actuals Request Structure above for the list of possible Answer Types

The Required Info Question Possible Values Level

TKRRequireInfoQuestionPssblVal (CNX_TKR_RQ_ANPV)

Under a Required Info Question entry, there can be 0, 1 or many Required Info Question Possible Value entries.

For Required Info entries whose answer type is from a list, the response contains the valid possible answer values.

The fields at the Required Info Question Possible Values level are:

- FIELDVALUE
 - The code associated with the question possible value
- DESCR100
 - The description associated with the question possible value

The Time Entered Response Structure

The Time Entered response structure is based on the PeopleSoft Message Container TKTimeEnteredResponse (CNX_TKI_TIME_ENTD_RESP_MC.V1) which is based on the PeopleSoft Message Part TKTimeEnteredResponse_MP (CNX_TKI_TIME_ENTD_RESP_MP.V1).

The Time Entered response structure is identical to Schedules and Actuals Response structure except the Time Entered response structure does NOT contain the Schedule / Actuals Level (TKTScheduleAndActualsResp).

- The Employee Level: TKSEmployeeResp (1)
 - The Date Under Report Level: TKDateUnderReportResp (1,M)
 - The Time Entered Level: TKTimeEnteredResp (0,1,M)
 - The Required Info Answers Level: TKTRequiredInfoAnswerResp (0,1,M)
 - The Time Reported Level: TKTimeReportedResp (0,1,M)
 - The Time Reported Messages Level: TKTimeReportedMessagesResp (0,1,M)
 - The Required Info Level: TKRRequiredInfoResp (0,1)
 - The Required Info Question Level: TKRRequiredInfoQuestionResp (1,M)

- The Required Info Answer Possible Values:
TKRRequireInfoQuestionPssblVal (0,1,M)

The yellow highlighted lines contain an “echo” of the request. The green highlighted lines contain the response from processing the request.

For more details about the various levels and the fields at each level, see the description for the Schedules and Actuals Response Structure above.

The Get Timekeeping State Response Structure

The Get Timekeeping State response structure is based on the PeopleSoft Message Container TKCurrentStateResponse (CNX_TKI_CURRENT_STATE_RESP_MC.V1) which is based on the PeopleSoft Message Part TKCurrentStateResponse_MP (CNX_TKI_CURRENT_STATE_RESP_MP.V1).

The Get TK State response structure contains a number of different levels, all of which contain data related to an employee’s Timekeeping State.

- The Employee Level: TKSEmployeeResp (1)
 - The Plan Balances as of Now Level: TKSPlanBalanceEELevelResp (0,1,M)
 - The Date Under Report Level: TKDateUnderReportResp (1,M)
 - The Job Level: TKSJobResp (1)
 - The Employee Name Level: TKSNamesResp (1)
 - The Time Reporter Level: TKSTimeReporterResp (0,1)
 - The Schedule Level: TKSScheduleResp (0,1)
 - The Plan Balances as of a DUR Level: TKSPlanBalancesResp (0,1,M)
 - The Time Entered Level: TKSTimeEnteredResp (0,1,M)
 - The Time Reported Level: TKSTimeReportedResp (0,1,M)
 - The Time Reported Messages Level: TKSTimeReportedMessagesResp (0,1,M)
 - The Reported Time Level: TKSReportedTimeResp (0,1,M)

The Employee Level

TKSEmployeeResp (CNX_TKS_EMPL)

There is exactly 1 Employee Level entry in each response.

- EMPLID
 - The PeopleSoft employee ID that uniquely identifies this employee in PeopleSoft
 - 6 digits
- EMPL_RCD
 - The employee record number that allows for the employee to have multiple relationships with the City of Winnipeg

- As we do not currently have “multiple jobs” in PeopleSoft, this will always be 0
- FROM_DATE
 - The earliest DUR for which the TK State was requested
 - Format: yyyy-mm-dd
- TO_DATE
 - The latest DUR for which the TK State was requested
 - Format: yyyy-mm-dd
- DTTM_CREATED
 - The date and time that the transaction was processed
 - Format: yyyy-mm-dd hh:mi:ss where hh is a 24 hour clock; all values will be padded on the left with zeroes

The Plan Balance as of Now Level

TKSPlanBalanceEELevelResp (CNX_TKS_PLN_BLE)

There are 0, 1 or many “Plan Balance as of Now” entries under the employee level.

For each compensatory time plan or each leave plan in which the employee is enrolled, this level holds the balance of the plan as of right now. This is different from an employee’s plan balance as of a DUR which is described below.

- CNX_PLAN
 - The Compensatory Time Off (CTO) (eg. OVERTIME) or Leave Plan (eg. 50 for Sick)
- CNX_COMP_OR_LEAV
 - A flag indicating if this Plan Balance is for a CTO Plan or a Leave Plan
 - C – CTO Plan
 - L – Leave Plan
- PLAN_DESCR
 - A description of the plan
- EFF_STATUS
 - The date on which the employee was most recently enrolled in the plan
- CNX_END_BALANCE
 - The balance in the plan as of right now as opposed to as of a certain DUR
 - See later for a description of the distinction between a plan balance as of right now and a plan balance as of a DUR

The Date Under Report Level

TKSDateUnderReportResp (CNX_TKS_DUR)

There are 1 or many Dates Under Report entries under the employee level.

- DUR
 - The date for which the transaction is being processed

- Format: yyyy-mm-dd

The Job Level

TKSJobResp (CNX_TKS_JOB)

There is exactly one Job entry under each DUR entry. The Job level holds information an employee's job data as of the DUR. Job information includes things about an employee as it relates to his job, such as hourly or salaried; standard hours; grade and step; pay rate; permanent vs temporary; part time vs full time; active vs suspend; etc.

There are too many fields at the Job level to list here. Most of the field names are pretty meaningful and the purpose of most fields will be understandable without further documentation.

The Name Level

TKSNamesResp (CNX_TKS_NAMES)

There is exactly one Names entry under each DUR entry. The Names level holds the employee name as of the DUR. There are currently quite a few different fields containing name information. After discussions with 311 and TKMMS, I will remove most of the fields. As a result, I will not be documenting all of the fields that currently exist in the Names level. I will document the few that I expect to stick around.

- NAME_DISPLAY
 - The way that the name will be displayed
- LAST_NAME_SRCH
 - A field that can be used to search by last name. All letters are capitalized.
- FIRST_NAME_SRCH
 - A field that can be used to search by first name. All letters are capitalized.
- LAST_NAME
 - The employee's last name in mixed case
- FIRST_NAME
 - The employee's first name in mixed case
- MIDDLE_NAME
 - The employee's middle name in mixed case

The Time Reporter Level

TKSTimeReporterResp (CNX_TKS_TM_RPTR)

There is exactly one Time Reporter entry under each DUR entry. The Time Reporter level holds information about an employee's Time Reporter status. This includes things like the time reporter status (Active or Inactive); time reporter type (Positive or Exception); workgroup; etc. These fields are mostly meaningful within Timekeeping and PeopleSoft and may not be relevant for external applications.

- EFFDT
 - The date on which this time reporter data came into effect
- TIME_RPTG_STATUS
 - The time reporting status
 - A time reporter must be active to report time
 - A – Active
 - I – Inactive
- TIME_RPTR_IND
 - Time reporter indicator
 - E – Elapsed (reports elapsed hours)
 - P – Punch (reports start and end times)
- WORKGROUP
 - The Time and Labour workgroup
- TASKGROUP
 - The Time and Labour taskgroup
- TASK_PROFILE_ID
 - The Time and Labour task profile
- TCD_GROUP_ID
 - The Time Collection Device group
- EFFDT_WRK
- EFF_STATUS
- TRC_PROGRAM
- RULE_PGM_ID
- TIME_RPTG_TYPE
 - The time reporter type
 - E – Exception (the employee only needs to report exceptions to his / her schedule)
 - P – Positive (the employee must report all time in order to get paid)
- HOLIDAY_SCHEDULE
 - What type of statutory holiday schedule is in effect for this employee
 - STD – Standard – statutory holidays may be moved to be observed on the first normal working day (Monday to Friday) after the date on which the statutory holiday falls
 - SHIFT – Shift – statutory holidays are observed on the day on which they fall
- CNX_IN_TK_WORKGRP
 - A Yes / No flag indicating if the employee is in a Timekeeping Self-Service workgroup
 - Y – The employee is in a TK Workgroup on the DUR in question
 - N – The employee is not in a TK Workgroup on the DUR in question

The Schedule Level

TKSScheduleResp (CNX_TKS_SCHD)

There are 0 or 1 Schedule Time entries under each DUR entry. The Schedule level holds information about an employee's schedule. This includes things like normal hours for the day, is the DUR a day off, is the DUR a day on which this employee observes a statutory holiday, etc. Employees who do not have an assigned schedule will not have an entry in the TK Schedule level.

- TL_QUANTITY
 - The number of regular working hours for the DUR
- WRKDAY_ID
 - The internal PeopleSoft identifier for the work day in effect on the DUR
- SCHEDULE_ID
 - The internal PeopleSoft identifier for the schedule in effect on the DUR
- SHIFT_ID
 - The internal PeopleSoft identifier for the shift in effect on the DUR
- OFFDAY_IND
 - A Yes / No flag indicating if the DUR is a day off
 - Y – Yes, it is a day off
 - N – No, it is not a day off
- SCHED_TYPE
 - An internal PeopleSoft identifier for the type of schedule in effect on the DUR
- TASKGROUP
 - An internal PeopleSoft identifier for the Task Group in effect on the DUR
- HOLIDAY_SCHEDULE
 - A code that identifies the statutory holiday schedule in effect for this employee for the DUR
 - STD – Standard – statutory holidays may be moved to be observed on the first normal working day (Monday to Friday) after the date on which the statutory holiday falls
 - SHIFT – Shift – statutory holidays are observed on the day on which they fall
- CNX_STAT_HOL_IND
 - A Yes / No flag indicating if a statutory holiday is observed on the DUR
 - Y – Yes, a statutory holiday is observed on that DUR
 - N – No, a statutory holiday is not observed on that DUR
- DESCR
 - The name of the statutory holiday if one is observed on that DUR

The Plan Balance as of a DUR Level

TKSPlanBalancesResp (CNX_TKS_PLN_BAL)

There are 0, 1 or many “Plan Balance as of a DUR” entries under the DUR level.

- CNX_PLAN
 - The Compensatory Time Off (CTO) (eg. OVERTIME) or Leave Plan (eg. 50 for Sick)

- CNX_COMP_OR_LEAV
 - A flag indicating if this Plan Balance is for a CTO Plan or a Leave Plan
 - C – CTO Plan
 - L – Leave Plan
- PLAN_DESCR
 - A description of the plan
- EFF_STATUS
 - The date on which the employee was most recently enrolled in the plan
- CNX_START_BALANCE
 - The balance in the plan as of the beginning of the DUR
- HOURS_EARNED
 - The number of hours earned in the plan on that DUR
- HRS_TAKEN
 - The number of hours taken from the plan on that DUR
- HRS_ADJ
 - The number of hours adjusted in the plan on that DUR
- CNX_END_BALANCE
 - The balance in the plan as of the end of the DUR
 - Calculation: Starting Balance + Earned + Adjusted – Taken
 - See later for a description of the distinction between a plan balance as of right now and a plan balance as of a DUR

The Time Entered Level

TKSTimeEnteredResp (CNX_TKS_TM_ENTD)

There are 0, 1 or many Time Entered entries under each DUR. The Time Entered level holds all of the existing Time Entered rows for an employee for the DUR.

Most fields are identical to the Time Entered level of the Time Entered Response structure. I will list all of the fields here but only put in additional notes if they are different from the notes above.

- SEQ_NBR
- CNX_TK_TC_CD
- CNX_TK_SC_CD
- CNX_TK_STATUS
- TL_QUANTITY
- CNX_TK_STC_GRP
- CNX_TK_SHIFT_BTN
- CNX_TK_PEER_BTN
- JOBCODE
- ACCT_CD
- CNX_TK_WCB_NBR

- CNX_TK_STC_FLD1
- CNX_TK_STC_FLD2
- CNX_TK_STC_FLD3
- CNX_TK_START_TIME
- CNX_TK_END_TIME
- CNX_TK_END_DTTM
- CNX_TK_START_DTTM
- CNX_TK_COMMENT
- TCD_ID
- CNX_TKS_STATE_STTS

The Time Reported Level

TKSTimeReportedResp (CNX_TKS_TM_RPTD)

Under the Time Entered entry, there can be 0, 1 or Many Time Reported entries.

The Time Reported level holds the “pre-reported” time that will ultimately get converted to PeopleSoft’s Reported Time.

Most fields are identical to the Time Reported level of the Time Entered Response structure. I will list all of the fields here but only put in additional notes if they are different from the notes above.

- ADDL_SEQ
- CNX_TK_TC_GRP
- CNX_TK_STATUS
- TRC
- TL_QUANTITY
- JOBCODE
- ACCT_CD
- OVERRIDE_RATE
- CNX_TK_COMMENT
- CNX_TK_DECEDENT
- CNX_TK_STC_ID
- CNX_TK_TC_ID
- USER_FIELD_1
- USER_FIELD_2
- USER_FIELD_3
- USER_FIELD_4
- USER_FIELD_5
- PROCESS_INSTANCE
- ADD_DELETE_IND
- PUBLISH_SW

- TCD_ID
- COMP_LEAV_IND
- CNX_PLAN
- CNX_COMP_OR_LEAV
- PLAN_DESCR
- FACTOR_MULT
- CNX_TKS_STATE_STTS

The Time Reported Messages Level

TKSTimeReportedMessagesResp (CNX_TKS_INT_TXN)

Under the Time Entered entry, there can be 0, 1 or Many Time Reported Message entries.

The Time Reported Messages Level can contain both

- Messages generated by the processing of a transaction
- The question and answer dialogue that is generated by the processing of a transaction

Most fields are identical to the Time Reported Messages level of the Time Entered Response structure. I will list all of the fields here but only put in additional notes if they are different from the notes above.

- ADDL_SEQ
- CNX_TKS_STATE_STTS
- CNX_TK_ANSWER
- CNX_TK_ANSWER_TYPE
- CNX_TK_REQ_INFO_ID
- CNX_TK_QUESTION

The Reported Time Level

TKSReportedTimeResp (CNX_TKS_RPTD_TM)

Under each DUR entry, there can be 0, 1 or Many Reported Time entries.

The Reported Time Level holds existing reported time (note the distinction between “reported time” which is the official PeopleSoft reported time (PS_TL_RPTD_TIME) and “time reported” which is the “pre-reported time” (PS_CNX_TK_TM_RPTD) generated by the Timekeeping application, that, once approved, is processed to become reported time.

The fields in the Reported Time level are all of the fields in the PS_TL_RPTD_TIME table with some additional fields appended that are used in the Timekeeping Interface. I will only list the additional fields here.

- COMP_LEAV_IND
 - How the time reported affects a plan

- CERN – Compensatory Time Off (CTO) Plan Earned
 - CTKN – CTO Taken
 - LTKN – Leave Taken
 - NO – No Plan affected
- CNX_PLAN
 - The specific compensatory time or leave plan affected
 - Eg. OVERTME, 50 (Sick), etc.
- CNX_COMP_OR_LEAV
 - A flag indicating if this time reported affects a Compensatory Time Plan (eg. overtime), a Leave Plan (eg. Vacation) or neither
 - C – Compensatory Plan
 - L – Leave Plan
 - Blank – no plan
- PLAN_DESCR
 - The description of the Plan affected
- FACTOR_MULT
 - The multiplication factor to be used for some reported time
 - For example, overtime that is paid at double-time will have a factor of 2.
- CNX_TKS_STATE_STTS
 - The Timekeeping State Status for this message
 - Possible values are LA – Logically Added and LD – Logically Deleted

The Get Employee ID Response Structure

The Get Timekeeping State response structure is based on the PeopleSoft Message Container TKGetEmplidResponse_MC (CNX_TKI_GET_EMPLID_RESP_MC.V1) which is based on the PeopleSoft Message Part TKGetEmplidResponse_MP (CNX_TKI_GET_EMPLID_RESP_MP.V1).

The request structure contains the following one-level hierarchy:

- The Employee Level: CNX_TKR_EMPLID (1)

The Employee Level

TKREmplid (CNX_TKR_EMPLID)

There will be 0 or 1 Employee Level entries in the response.

- EMPLID
 - The employee ID associated with the Active Directory ID sent in the request
- EMPL_RCD
 - The employee record number associated with the Active Directory ID sent in the request
 - As we do not currently have “multiple jobs” in PeopleSoft, this will always be 0

The Response Structure and the User Specified Response Level

The following applies to the response that is returned from the services CNX_TKI_PROCESS_SCHD_ACTUALS and CNX_TKI_PROCESS_TIME_ENTERED.

As documented above, the requester of these two services can specify the level of detail that should be returned in the response for these services. See the request structure and the field CNX_TK_RESP_LEVEL under “The Employee Level” for each of these services.

There are actually three main variables that determine which levels and fields will be returned in the response to one of these services:

- The response level of detail requested
 - There are three levels of detail which can be requested
 - L – Low – show only the success flags
 - M – Medium – Low plus the time entered, time reported and time reported messages that result from the processing of the transaction
 - H – High – Medium plus “echo” back the Schedules and Actuals transaction if the request is of type Schedule and Actuals
- The success of the transaction or part of the transaction
 - If a Low or Medium response level is requested and a sub-transaction fails, it is not enough to return a response flag of N. The system must also return enough details so that the caller can figure out why the transaction failed.
 - For example, if the transaction failed because not all required information was provided, the response needs to show the details of the request and the specific Required Info for which an answer was not provided (and could not be derived)
- The level at which the database is being updated
 - The three levels at which the database can be updated are:
 - At the “time entered” level
 - As each time entered sub-transaction is successfully processed, the database is updated
 - This is how the self-service application currently works
 - At the DUR level
 - Once all sub-transactions for a DUR have been successfully processed, the database is updated
 - If any sub-transaction for a DUR fails, no sub-transactions for that DUR are applied to the database and no subsequent DURs are processed
 - At the complete transaction level
 - Once all of the sub-transactions for all DURs have been successfully processed, the database is updated
 - If any sub-transaction for any DUR fails, no sub-transactions for the entire transaction are applied to the database

Replacing Data for the DUR

The following applies to the services CNX_TKI_PROCESS_SCHD_ACTUALS and CNX_TKI_PROCESS_TIME_ENTERED.

When the Timekeeping Interface processes a Schedule / Actuals transaction or a Time Entered transaction, it will delete all existing pre-reported time that came from the external system sending the transaction. That is, the TKI is expecting the external system to send a complete set of data for the employee for each DUR for which time is to be processed.

For example, if the external system first sends 4 hours of regular time and 3 hours of sick time and then later it is determined that the employee was on vacation and not sick, then the second transaction for this employee for this DUR should include both the 4 hours of regular time and the 3 hours of vacation time.

The TKI will logically delete all pre-reported time for an employee for a DUR covered by a transaction and then logically add in the new information. If the transaction completes successfully and the database is requested to be updated, then these logical deletes and adds will be physically made to the database.

Note that if data were incorrectly sent for an employee for a DUR and this time should be deleted, the external system simply needs to send a transaction for that employee for the DUR in question with no Schedule / Actuals or Time Entered information for that DUR. The TKI will logically delete the existing time and if the transaction is successful, physically delete the time.

Generic Time Codes, Subcodes and Required Info Ids

The following is a complete list of the current Generic Time Codes (GTC) that are used in the TKI.

For each GTC, if there are subcodes, then they are listed as well.

For each GTC, the Required Info Ids (ie. the internal Timekeeping codes indicating a need for more information) are also listed. Some required Info Ids have answers that can be derived based on Schedule and Actuals information provided. Required Info Ids whose answers cannot be derived from the Schedules and Actuals must have the answer values provided as part of the transaction and are noted with an asterisk.

ANP - Away Without Pay

- Subcodes
 - DAB - Disciplinary
 - EAB - Excused Absence
 - LAB - Lateness
 - NAB - Non-Disciplinary
 - UAB - Unexcused Absence

- Required Info Ids
 - None

AWP - Away With Pay

- Subcodes
 - DAB - Disciplinary
 - EAB - Excused Absence
 - LAB - Lateness
 - SVA - Vacation Sick
 - UNB - Union Business
- Required Info Ids
 - None

BTC - Banked Time Cash Out

- Subcodes
 - LSR - Long Service Recognition
 - OTR - Overtime
 - SPR - Shift Premium
 - STB - Standby
 - STH - Statutory
 - STL - Stat Lieu
 - WWL - Water & Waste Lieu
- Required Info Ids
 - None

BTT - Bank Time Off

- Subcodes
 - CMP - Compressed
 - OTR - Overtime
 - SPR - Shift Premium
 - STB - Standby
 - STH - Statutory
 - STL - Stat Lieu
 - WWC - Water & Waste Comp Time
 - WWL - Water & Waste Lieu
- Required Info Ids
 - None

CPW - Compressed Work Hrs To Bank

- Subcodes
 - None
 - Required Info Ids
-

- None

CTM - Card Time

- Subcodes
 - None
- Required Info Ids
 - None

DIS - Disinterment

- Subcodes
 - None
- Required Info Ids
 - isGrave50YrsOld*
 - "Is the grave more than 50 years old?"
 - YesNo
 - isTimeInRegShift
 - "Was this work done during your regular shift?"
 - YesNo
 - StartTimeofWork
 - "What time did you start this work?"
 - Time
 - EndTimeofWork
 - "What time did you end this work?"
 - Time

EDU - Education and Training

- Subcodes
 - REI - Reimbursement
 - PAD - On a Course
 - WWE - Water/Waste Educ & Training
- Required Info Ids
 - REI: AmountGST*
 - "How much, if any, of the amount was GST? If there was no GST, you must enter 0."
 - Number
 - WWE: AnsDayOff (for W&W)
 - "Was this your day off?"
 - YesNo
 - WWE: BankThisTime (for W&W)*
 - "Do you want to bank your time?"
 - YesNo

FLD - Flood

- Subcodes
 - OTR - Overtime
 - REG - Regular
 - STB - Standby
 - STH - Statutory
 - TSB - Transit Standby
- Required Info Ids
 - IncrementalPay*
 - “Is this incremental pay?”
 - YesNo
 - StartDate
 - ?
 - ?

GRR - Gradual Returns

- Subcodes
 - DSP - Disability Paid
 - DSU - Disability Unpaid
 - SAD - Sick Alternate Duties
 - WAD - Workers Comp Alternate Duties
 - WCR - Workers Comp Rehab
 - WMD - Workers Comp Modified Duties
- Required Info Ids
 - Workers Comp related: HSClaimNbr*
 - “What is the related WCB Claim Number?”
 - FromList

LEV - Leaves

- Subcodes
 - BLV - Bereavement
 - CLV - Compassionate
 - FAM - Family
 - FLV - Furlough
 - FRL - Funeral
 - JRY - Jury/Court
- Required Info Ids
 - BLV: WhoPassedAway*
 - “Who has passed away?”
 - FromList
 - FRL: PallbearerAtFuneral*

- “Were you a pallbearer at this funeral?”
 - YesNo
- JRY: ProofOfJuryService*
 - “Did you give your supervisor proof of jury service or your subpoena for being a witness?”
 - YesNo
- NotEnoughSickCredits*
 - Not sure if used by Leaves GTC

MUD – Make Up Day (related to Sent Home)

- Subcodes
 - None
- Required Info Ids
 - MakeUpDates*
 - "What dates are you making time up for?"
 - Date
 - HoursMissed*
 - "How many hours did you miss because of inclement weather or equipment breakdown? NOTE: This includes any vacation or banked time that you used on that day due to being sent home."
 - Number
 - Plus all Required Info Ids associated with ROT – Regular Overtime

REG - Regular

- Subcodes
 - None
- Required Info Ids
 - None

ROT - Overtime

- Subcodes
 - None
- Required Info Ids
 - WhenWasOT
 - "When did you work your overtime?"
 - FromList
 - isGapAfterShift
 - "Was there more than a 2 hour gap between when your shift finished and overtime started?"
 - YesNo
 - isOTDuringNightHours

- "Were any of your overtime hours between midnight and 8:00 am?"
 - YesNo
- OTStartTime
 - "When did you start your overtime?"
 - Time
- RegShiftStartTime
 - "When did you start your shift?"
 - Time
- RegShiftEndTime
 - "When did you end your shift?"
 - Time
- isSuperProvidedSupper*
 - "Did your supervisor provide supper?"
 - YesNo
- isNotifiedOfOT*
 - "Were you notified of the overtime before you went home?"
 - YesNo
- isWorkingStandby
 - "Were you on Standby at the time of this Overtime?"
 - YesNo
- isDayOff
 - "Did you work the overtime on your day off?"
 - YesNo
- isBankOT*
 - "Did you want to bank your overtime?"
 - YesNo
- isEmergencySituation*
 - "Did you report to work as a result of an emergency situation?"
- BankTheOvertime
 - "Do you want to bank your " | %This.Remainder | " hours?"
 - YesNo

SIC - Sick

- Subcodes
 - None
- Required Info Ids
 - wasMedicalAppointment*
 - "Was this for a Medical Appointment in Winnipeg?"
 - YesNo
 - haveDoctorsNote*
 - "Did you provide a Doctor's note to your Supervisor?"

- YesNo
- duetoIllnessInFamily*
 - "Is time off due to illness in family?"
 - YesNo
- NotEnoughSickCredits*
 - "You do not have enough Sick Credits to cover " | NumberToString("%7.2v", &Amount) | " hours of this time. Do you have any other Credits you would like to use? "
 - YesNo
- NotEnoughSickFamilyCredits*
 - "You do not have enough Sick Family Credits to cover " | NumberToString("%7.2v", &Amount) | " hours of this time. Do you have any other Credits you would like to use? "
 - YesNo

SNT - Sent Home - Weather/Breakdown

- Subcodes
 - None
- Required Info Ids
 - ReportForWork (non-PARKS)*
 - "Did you report for work today?"
 - YesNo
 - HoursWorked (non-PARKS)
 - "How many hours did you work after reporting? Note that your answer can be 0."
 - Number
 - isGivenOptionToWork (PARKS)*
 - "Were you given the option to work your shift?"
 - YesNo
 - isUseVacationOrBank (PARKS)*
 - "If you have vacation or banked time, you have the option of using that time instead of not getting paid. Are you going to use this option?"
 - YesNo

SOT - Special Overtime

- Subcodes
 - None
- Required Info Ids
 - HoursWorked
 - "How many hours did you work today in your regular shift?"
 - Number

- RegShiftStartTime
 - "What time did your regular shift start today? "
 - Time
- PrevOverTimeWorked
 - "Did your last overtime worked prior to this regular shift end before " | This.RegStartDateTime?"
 - YesNo
- WorkedMoreThan4Hrs
 - "Did you work more than 4 hours during this overtime? "
 - YesNo
- OverTimeStartYesterday
 - "Did this overtime start yesterday? "

STB - Standby

- Subcodes
 - None
- Required Info Ids
 - SBStartTime
 - "Standby Start Time : "
 - Time
 - BankThisTime*
 - "Do you want to bank this time?" (CUPE)
 - "Do you want to bank these hours you were on standby?" (WAPSO)
 - YesNo
 - ShiftForSomeoneElse*
 - "Did you take over for someone else during this standby shift?"
 - YesNo

STH - Statutory

- Subcodes
 - None
- Required Info Ids
 - HowManyHoursWorked
 - "How many hours did you actually work?"
 - Number
 - isBankStatTime*
 - "Do you want to bank the time that you worked on the statutory holiday?"
 - YesNo
 - isDayOff
 - "Did you work the overtime on your day off?"
 - YesNo

- isNormalDayOfWork
 - "Is this your normal day of work?"
 - YesNo
- NormalHoursIfNotStat
 - "How many hours would you have worked if this day were not a stat holiday?"
 - Number
- isDidWorkStat
 - "Did you work on the stat holiday?"
 - YesNo
- isLastWorkingDayBeforeXmas
 - "Is this your last normal working day before " &dt_LastWorkingDayBeforeXmas?"
 - YesNo
- NormalHoursIfNotXmasEve
 - "How many hours would you have worked if this day were not Christmas Eve?"
 - Number
- Plus all Required Info Ids associated with ROT – Regular Overtime

TSB - Transit Standby

- Subcodes
 - None
- Required Info Ids
 - SBRotationStartDate
 - "What day did you start your standby rotation? "
 - Date
 - LastDayOfRotation
 - "Is today the last day of your standby rotation? "
 - YesNo

VAC - Vacation

- Subcodes
 - None
- Required Info Ids
 - AdvanceApproval*
 - "You do not have enough vacation credits to cover this time off. Do you have written approval for advance vacation?"
 - YesNo
 - DifferentCredits*
 - "Do you have credits of a different type you wish to use for the remaining time?"
 - YesNo
 - NoPayPermission*

- "Do you have permission to take this vacation request at no pay?"
- YesNo

VWL - Vacation/WAPSO Days/Long Serv.

- Subcodes
 - None
- Required Info Ids
 - AdvanceApproval*
 - "You do not have enough vacation credits to cover this time off. Do you have written approval for advance vacation?"
 - YesNo
 - DifferentCredits*
 - "Do you have credits of a different type you wish to use for the remaining time?"
 - YesNo
 - NoPayPermission*
 - "Do you have permission to take this vacation request at no pay?"
 - YesNo
 - UseLongService*
 - "You have " | &LeavePlanBal | " hrs left in Long Service Recognition bonus that you must use by December 31st, " | &LspYear | ". Do you want to use " | &temp | " hrs now?"
 - YesNo

WST - Work Stoppage

- Subcodes
 - OTR - Overtime
 - REG - Regular
 - SIC - Sick
 - STB - Standby
- Required Info Ids
 - SIC: isSickRelatedToWS*
 - "Was your sickness related to the work stoppage?"
 - YesNo
 - SIC: isDrNoteSupplied*
 - "Did you supply a doctor's note to your supervisor?"
 - YesNo
 - SIC: isDrNoteApproved*
 - "Was the doctor's note approved by the City's medical practitioner or designate?"
 - YesNo

- STB: BankThisTime*
 - "Do you want to bank this time?"
 - YesNo
- Plus all Required Info Ids associated with
 - ROT – Regular Overtime
 - REG – Regular
 - SIC – Sick
 - STB – Standby