**This document and its content is copyrighted by Kingland Systems Corporation and may not be copied, shared with others, or used for purposes other than intended, without the express written permission of Kingland Systems Corporation.  All rights reserved.**

# Introduction

Since 360 View is an already existed component so what we need to do is enhance it and make it fulfill the new requirement. In the old 360 View component, it can only display the current view of an organization or Security. In the new component, we need to be able to display the snapshot of a record in the past.

# User Interface Design

For the screenshot of 360 View-Orgnization(Details Page),please see the 6.1 part of the requirement-291433. For Security, please see 7.1.

For special requirement, please refer to the section 3, 4, 6.6 of the requirement.

## Component Name

Org 360 View - Details

### Views and Controllers

**com.kingland.clarity.client.organization.threesixtyview.DetailsPanel(new class)**

1. Create the above new class(It will contain the content of the details tab of Org360View): public class DetailsPanel extends Composite implements Content.

2. Copy the existing Org360View.class’s content into the above new class.

Then in the new class:

3. Delete the Tree(setupTreeItems() method) because we don’t need it now; Delete all of the methods which are used for getting the stage data. Like setupLists(), getStageBaseDetailsPanel(), getStageNamesPanel()... In the new 360 View, we don't need to display the stage data.

4. Add one incoming parameter into the constructor method of the class. Because we will get the initial Organization object from the main class **Org360View** and display its content on the screen as the default page. So **Org360View** will pass us the default Organization object. Therefor the type of the new incoming object will be TCRMOrganizationBObjProxy.

5. Create a new method called: getViewAsOfContainer(). It will help us generate the first panel of the screen. In this method:

Create a new Combobox and its data is from field "H\_END\_DT" of "H\_CONTACT" table.

In OrganizationService.class, call the below method “**GetEndDateHistoryFromContactHistory**” which is in the “Service Interface” section below. If the value we get from the response is null or blank, set the Combobox's actual value as the current timestamp(Format : "yyyy-MM-dd HH:mm:ss.SSS"). Sample code:

Date currentTime = **new**Date();

DateTimeFormatfmt = DateTimeFormat.*getFormat*("yyyy-MM-ddHH:mm:ss.SSS");

String dateString = fmt.format(currentTime);

HistoryOfEndDateBObjProxy.setHistoryEndDate(dateString);

If not, reduce the returned value by 1.5 second and put the time into the Combobox(In Organization, we found that the timestamp of every table's H\_END\_DT might be disordered. It means that some times the H\_END\_DT of H\_CONTACT might even be later than Org table's H\_CREATE\_DT. So we use 1.5 seconds as the buffer time). Below is the sample code:

DateTimeFormatfmt = DateTimeFormat.*getFormat*("yyyy-MM-ddHH:mm:ss.SSS");

Date date = fmt.parse(response.getData().get(i).getHistoryEndDate());

date = **new** Date(date.getTime() - 1500);

HistoryOfEndDateBObjProxy.setHistoryEndDate(fmt.format(date));

Then we can start taking care of the displayed value of the combo box since requirement said we need to show “Current” for the latest record and timestamp with “AM” or “PM” for other records. Below is the sample code:

Combo = **new**ComboBox<HistoryOfEndDateBObjProxy>(Store, **new**LabelProvider<HistoryOfEndDateBObjProxy>()

{

@Override

**public** String getLabel(HistoryOfEndDateBObjProxyitem)

{

**if** (CurrentTime.equals(item.getHistoryEndDate())){

**return**"Current";

} **else** {

KSCDateStringFormatterkSCDateStringFormatter =**new**KSCDateStringFormatter(DateTimeFormat.*getFormat*("yyyy-MM-ddHH:mm:ss.SSS"), DateTimeFormat.*getFormat*("yyyy-MM-ddhh:mm:ss a"));

**return**kSCDateStringFormatter.format(item.getHistoryEndDate());

                  }

}

});

Put this Combobox and its front lable "View as of " on the right side of a HorizontalLayoutContainer and return this HorizontalLayoutContainer back.

Add a change event to the Combobox. Every time when customer changes his option, get the value customer chose and send it to MDM as well as the Organization ID(Refer to the Service Interface section of this document - **SearchOrganizationHistoryDetail**). Then display the returned result on the screen. Below are something you need to pay attention to:

1. In **SearchOrganizationHistoryDetail**, we will call the MDM side’s inquiry “getOrganization” and pass the “As Of Date” in so we can get the history records. But the current inquiry provided by MDM has something needed to be paid attention to. For example, if you delete an organization name, you can see it returns the left organization name several times instead of once, so we have duplicate records, I think this is MDM’s design and for now we can make a logic to ignore the duplicate returned values and just display one of them. You can get the returned organization name at first and then check whether its “EndDate” is null or after the current time, and only displays the records which have the null “EndDate” or “EndDate” is after the current time – Note: This “EndDate” is not the “H\_END\_DT” and do the same for the other necessary attributes.

2. The Identifier record with the type of “Hub ID” will not be displayed in the “Identifier” table. So we need to create some logics to make this happen. Since the type “Hub ID” corresponding to Identification Type “1000” in Organization so we need to retrieve all identifiers out and then check their Identification Type, if it’s not “1000”, add this record to the Identifier table.

6. Create a new panel named “Child Relationships”. Refer to the way “Parent Relationships” generated. Its value should come from “loadData()” method: if TCRMPartyRelationshipBObjProxy.getRelationshipFromPartyId() equals to Org’s party id, add this record to the panel.

7. In buildUI() method, create a new VerticalLayoutContainer and add the Container getViewAsOfContainer() returned back into it. And then also add the other panels in. Like the panel returned by getBaseDetailsPanel(), getNamesPanel()…Please be aware that these panels need to be ContentPanel and we need to set their Collapsible attribute to true so customer can collapse and expand the table. Also the ContentPanel will show all records within the table without a vertical scrollbar automatically. And for every store of these grids, use “addSortInfo” to set their default order as Last Update Date in Descending order.

At the last of buildUI(), load all of the current data in: We need to get the content of TCRMOrganizationBObjProxy passed by **Org360View**and then display the result on the screen. Below are something you need to pay attention to:

1. In **SearchOrganizationHistoryDetail**, we will call the MDM side’s inquiry “getOrganization” and pass the “As Of Date” in so we can get the history records. But the current inquiry provided by MDM has something needed to be paid attention to. For example, if you delete an organization name, you can see it returns the left organization name several times instead of once, so we have duplicate records, I think this is MDM’s design and for now we can make a logic to ignore the duplicate returned values and just display one of them. You can get the returned organization name at first and then check whether its “EndDate” is null or after the current time, and only displays the records which have the null “EndDate” or “EndDate” is after the current time – Note: This “EndDate” is not the “H\_END\_DT” and do the same for the other necessary attributes.

2. The Identifier record with the type of “Hub ID” will not be displayed in the “Identifier” table so we need to create some logics to make this happen. Since the type “Hub ID” corresponding to Identification Type “1000” in Organization so we need to retrieve all identifiers out and then check their Identification Type, if it’s not “1000”, add this record to the Identifier table.

8. Create a new overwrite method: public Widget getContent(). In it, return the above VerticalLayoutContainer.

**com.kingland.clarity.client.organization.threesixtyview.LineagePanel(new class)**

Copy the existing Org360View.class’s content into the above new class. Change the above class a little so it can be displayed on a tab. The layout doesn’t need to change.

**com.kingland.clarity.client.organization.threesixtyview.Org360View**

The above class will be the main class to display our 3 tabs. It will get the content of 3 tabs and put them on a window. Below is the general instruction:

At first, use below code to make the height and width of the 360 View window size to allow 100 pixels of space between the edge of the user's screen and the window on all sides:

setPosition(10,10);

setSize(String.*valueOf*(XDOM.*getViewportWidth*() - 20),String.*valueOf*(XDOM.*getViewportHeight*() - 20));

Then, prepare data for the title and details tab: Use “**SearchOrganizationHistoryDetail**” method and pass the current time in as the “As Of Date”. Also pass in the Organization id. After we get the result, at first, we need to make the title for our page. Below are things you need to pay attention to:

1. We need to retrieve the “Display Order” ‘s name out and make it part of the page’s title. For how to get “Display Order”, this is regarding to the Code Table Improvements and is still in the progress of design. You can refer to task 280967 for details.

2. The Identifier record with the type of “Hub ID” will not be displayed in the “Identifier” table but its ID will be part of our page’s title. So we need to create some logics to make this happen. Since the type “Hub ID” corresponding to Identification Type “1000” in Organization so we need to retrieve all identifiers out and then check their Identification Type, if it’s “1000”, put this record’s id as part of the page title.

3. Call **setHeadingText()** method. The title of the 360 View will display the search icon    followed by a hypen, then the first active name of the Organization (or Security) based on the sort priority order, followed by a hypen and the Hub Id(Will be the identifier of type Hub Id) of the record.

4. Pass the object we get from “**SearchOrganizationHistoryDetail**” to DetailsPanel.class and put the returned VerticalLayoutContainer by DetailsPanel.class into a PlainTabPanel and put the PlainTabPanel into the BorderLayoutContainer. Then call add(BorderLayoutContainer) method.

## Component Name

Sec 360 View - Details

### Views and Controllers

**com.kingland.clarity.client.security.threesixtyview.DetailsPanel(new class)**

Refer to the change in DetailsPanel.class. We need to call “**GetEndDateHistoryFromSecurityHistory**” to get the ViewAsOf field’s value. In Security, we don’t need to make logic to ignore the duplicate records like Organization Name. And the type “Hub ID” corresponding to Identification Type “1020” in Security.

**com.kingland.clarity.client.security.threesixtyview.LineagePanel(new class)**

The layout should be the same as the old Security360View.class.

**com.kingland.clarity.client.security.threesixtyview.Security360View**

Refer to the change in Org360View.class.

# Service Interface

## Component Name

Org 360 View - Details

### Service contract updates

**com.kingland.hub.server.data.rmi.GetEndDateHistoryFromContactHistory (new class)**

**GetEndDateHistoryFromContactHistory extends MDMCommandTemplate**

This class is for getting the content of our combobox on the page. It will call MDM and execute a SQL to query our main history table to see how many times the record is updated and when the update occurred. For Organization, it queries H\_CONTACT table and for Security, it queries H\_KSCSECURITY table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Contracts** | | | |
| Method Name | Method Parameters | Return Parameters | Comment |
| public GetEndDateHistoryFromContactHistory() throws NamingException | None | None | Call the super(null) and initiate the ArrayList<HistoryOfEndDateBObj> |
| public String getStylesheetName() | None | String | Return the xml file: GetEndDateHistoryFromContactHistory.xsl |
| public Map<String, Object> getStylesheetParameters(String request) throws Exception | String | Map<String, Object> | Put the id we get from the front end into a map. |
| public List<HistoryOfEndDateBObj> processResults(Document result) throws Exception | Document | List<HistoryOfEndDateBObj> | Get the return result from MDM. |
| public void processFailure(DWLResponseException failure) throws Exception | DWLResponseException | void | Exception handling. |
| public Map<String, Object> getStylesheetParameters(List< String > request) throws Exception | List< String > | Map<String, Object> | Throw an exception in it since this method should not be called. |
| public HistoryOfEndDateBObj processResult(Document result) throws Exception | Document | HistoryOfEndDateBObj | Throw an exception in it since this method should not be called. |

**com.kingland.hub.server.data.model.HistoryOfEndDateBObj(new class)**

|  |  |  |
| --- | --- | --- |
| **HistoryOfEndDateBObj** | | |
| Name | Type | Comment |
| **historyEndDate** | String |  |

**com.kingland.hub.server.data.rmi.SearchOrganizationHistoryDetail(new class)**

**SearchOrganizationHistoryDetail extends MDMCommandTemplate**

This method will be used to call the MDM side’s inquiry “getOrganization”. If you take a look at the old Org360View.class, you can find a method called “getOrganizationById”. You can see it also calls MDM side’s inquiry “getOrganization”. But since it just has one parameter for the id and doesn’t have one for the “As Of Date” so we need to new a object to store the Organization ID and the “As Of Date” customer chose from the combobox. Please see the below object “**SearchOrganizationHistoryDetailInBObj**”.

And the below method needs a xml template so it can exchange information with MDM. So we need to new a xml template. Please see the template in “XML Structures” section. Please note that it calls the “getOrganization” inquiry and it adds a new parameter “inquireAsOfDate” in the DWLCONTROL. So in the below method, we need to put the value we get from “**SearchOrganizationHistoryDetailInBObj**” in the corresponding filed in the xml template. Below are sample codes in “getStylesheetParameters” method:

HashMap<String, Object> params = **new** HashMap<String, Object>();

params.put("id", request.getPartyId());

params.put("inquireAsOfDate", request.getInquireToDate());

**return** params;

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Contracts** | | | |
| Method Name | Method Parameters | Return Parameters | Comment |
| public SearchOrganizationHistoryDetail() | None | None | Call the super(null) and initiate the TCRMOrganizationBObj |
| public String getStylesheetName() | None | String | Return the xml file: SearchOrganizationHistoryDetail.xsl |
| public Map<String, Object> getStylesheetParameters(SearchOrganizationHistoryDetailInBObj request) throws Exception | SearchOrganizationHistoryDetailInBObj | Map<String, Object> | Put the parameters into a map. |
| public TCRMOrganizationBObj processResult(Document result) throws Exception | Document | TCRMOrganizationBObj | Get the return result from MDM. |
| public void processFailure(DWLResponseException failure) throws Exception | DWLResponseException | void | Exception handling. |
| public Map<String, Object> getStylesheetParameters(List<SearchOrganizationHistoryDetailInBObj> request) throws Exception | List< SearchOrganizationHistoryDetailInBObj> | Map<String, Object> | Throw an exception in it since this method should not be called. |
| public List<TCRMOrganizationBObj> processResults(Document result) throws Exception | Document | List< TCRMOrganizationBObj> | Throw an exception in it since this method should not be called. |
| public void setIndustryTypeJDBC(IndustryTypeJDBC industryTypeJDBC) | IndustryTypeJDBC | void | Handle industry type. |
| public IndustryTypeJDBC getIndustryTypeJDBC() | None | IndustryTypeJDBC | Handle industry type. |

**com.kingland.hub.server.data.model.SearchOrganizationHistoryDetailInBObj(new class)**

|  |  |  |
| --- | --- | --- |
| **SearchOrganizationHistoryDetailInBObj** | | |
| Name | Type | Comment |
| **partyId** | String |  |
| **asOfDate** | String |  |

## Component Name

Sec 360 View - Details

### Service contract updates

**com.kingland.hub.server.data.rmi.GetEndDateHistoryFromSecurityHistory (new class)**

**GetEndDateHistoryFromSecurityHistory extends MDMCommandTemplate**

This class is for getting the content of our combobox on the page. It will call MDM and execute a SQL to query our main history table to see how many times the record is updated and when the update occurred. For Organization, it queries H\_CONTACT table and for Security, it queries H\_KSCSECURITY table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Contracts** | | | |
| Method Name | Method Parameters | Return Parameters | Comment |
| public GetEndDateHistoryFromSecurityHistory() throws NamingException | None | None | Call the super(null) and initiate the ArrayList<HistoryOfEndDateBObj> |
| public String getStylesheetName() | None | String | Return the xml file: GetEndDateHistoryFromSecurityHistory.xsl |
| public Map<String, Object> getStylesheetParameters(String request) throws Exception | String | Map<String, Object> | Put the id we get from the front end into a map. |
| public List<HistoryOfEndDateBObj> processResults(Document result) throws Exception | Document | List<HistoryOfEndDateBObj> | Get the return result from MDM. |
| public void processFailure(DWLResponseException failure) throws Exception | DWLResponseException | void | Exception handling. |
| public Map<String, Object> getStylesheetParameters(List<String> request) throws Exception | List<String> | Map<String, Object> | Throw an exception in it since this method should not be called. |
| public HistoryOfEndDateBObj processResult(Document result) throws Exception | Document | HistoryOfEndDateBObj | Throw an exception in it since this method should not be called. |

**com.kingland.hub.server.data.model.HistoryOfEndDateBObj(Same one as the one in Organization part)**

|  |  |  |
| --- | --- | --- |
| **HistoryOfEndDateBObj(The same object as the one in Organization part)** | | |
| Name | Type | Comment |
| **historyEndDate** | String |  |

**com.kingland.hub.server.data.rmi.SearchSecurityHistoryDetail (new class)**

**SearchSecurityHistoryDetail extends MDMCommandTemplate**

The same use as the Organization part. Need to call SearchSecurityHistoryDetail.xsl.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Contracts** | | | |
| Method Name | Method Parameters | Return Parameters | Comment |
| public SearchSecurityHistoryDetail() | None | None | Similar as Organization part. |
| public String getStylesheetName() | None | String | Return the xml file: SearchSecurityHistoryDetail.xsl |
| public Map<String, Object> getStylesheetParameters(SearchSecurityHistoryDetailInBObj request) | SearchSecurityHistoryDetailInBObj | Map<String, Object> | Similar as Organization part. |
| public KSCSecurityBObj processResult(Document result) | Document | KSCSecurityBObj | Similar as Organization part. |
| public void processFailure(DWLResponseException failure) | DWLResponseException | void | Similar as Organization part. |
| public Map<String, Object> getStylesheetParameters(List< SearchSecurityHistoryDetailInBObj> request) | List< SearchSecurityHistoryDetailInBObj> | Map<String, Object> | Similar as Organization part. |
| public List< KSCSecurityBObj> processResults(Document result) | Document | List< KSCSecurityBObj> | Similar as Organization part. |

**com.kingland.hub.server.data.model.SearchSecurityHistoryDetailInBObj(new class)**

|  |  |  |
| --- | --- | --- |
| **SearchSecurityHistoryDetailInBObj** | | |
| Name | Type | Comment |
| **id** | String |  |
| **asOfDate** | String |  |

# Business Logic

## Component Name

Org 360 View - Details

### Class Name (repeat for each class)

***KinglandOrg***

**module.mdmxmi**

**New a folder against KinglandOrg, called ContactHistory. Then in ContactHistory folder, add a new transient object:**

**HistoryOfEndDate: HistoryOfEndDate has 1 String attribute: historyEndDatestores the value we get from the DB.**

We need to new an inquiry Transaction for retrieving the history end dates from history tables. New an inquiry Transaction called **GetEndDateHistoryFromContactHistory**. Add a String parameter to it called “partyId”. Its response type: **HistoryOfEndDate**. Generate the code then.

**ContactHistoryComponent.class**

In handleGetEndDateHistoryFromContactHistory method, new a QueryConnection and execute the SQL: "select H\_END\_DT from H\_CONTACT where H\_CONT\_ID=? order by H\_END\_DT DESC". Sample code:

QueryConnection connection = **null**;

ListsqlParam = **new**ArrayList();

sqlParam.add(partyId);

ResultSetrs = **null**;

connection = DataManager.*getInstance*().getQueryConnection();

String myQuery = "select H\_END\_DT from H\_CONTACT where H\_CONT\_ID=? order by H\_END\_DT DESC";

**if** (connection !=**null**)

   rs = connection.queryResults(myQuery, sqlParam.toArray());

Then return the result back.

## Component Name

Sec 360 View - Details

### Class Name (repeat for each class)

***KinglandSecurity***

**module.mdmxmi**

In KinglandSecurity folder, create an inquiry Transaction called GetEndDateHistoryFromSecurityHistory for retrieving history End Date from history tables. Create a String parameter for it called “securityId”. Its response type is HistoryOfEndDate. Generate the code then.

**KinglandSecurityComponent.class**

In handleGetEndDateHistoryFromSecurityHistory method,  new a QueryConnection and execute the SQL: "select h\_end\_dt FROM H\_KSCSECURITY WHERE H\_id = ? order by h\_end\_dt DESC". Sample code:

QueryConnection connection = **null**;

ListsqlParam = **new**ArrayList();

sqlParam.add(securityId);

ResultSetrs = **null**;

connection = DataManager.*getInstance*().getQueryConnection();

String myQuery = "select h\_end\_dt FROM H\_KSCSECURITY WHERE H\_id= ?order by h\_end\_dt DESC";

**if** (connection != **null**)

   rs = connection.queryResults(myQuery, sqlParam.toArray());

Then return the result back.

# XML/File Structures

Sample xml for getting the End Date History value from Contact History table(GetEndDateHistoryFromContactHistory.xsl), need to be called by GetEndDateHistoryFromContactHistory.getStylesheetName method:

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!--

Stylesheet that creates a getPerson XML file that can be sent to the MDM Server.

 The information that the stylesheet needs is all passed in as parameters.

  -->

<xsl:stylesheetxmlns:xsl=*"http://www.w3.org/1999/XSL/Transform"*version=*"1.0"*>

   <xsl:paramname=*"requestId"*/>

   <xsl:paramname=*"requesterName"*/>

   <xsl:paramname=*"requesterLanguage"*/>

   <xsl:paramname=*"partyId"*/>

   <xsl:outputmethod=*"xml"*/>

   <xsl:templatematch=*"/"*>

      <TCRMServicexmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xsi:noNamespaceSchemaLocation=*"KSCRequest.xsd"*>

          <RequestControl>

              <requestID><xsl:value-ofselect=*"$requestId"*/></requestID>

              <DWLControl>

                  <requesterName><xsl:value-ofselect=*"$requesterName"*/></requesterName>

                  <requesterLanguage><xsl:value-ofselect=*"$requesterLanguage"*/></requesterLanguage>

              </DWLControl>

          </RequestControl>

          <TCRMInquiry>

              <InquiryType>GetEndDateHistoryFromContactHistory</InquiryType>

              <InquiryParam>

                 <tcrmParamname=*"*partyId*"*><xsl:value-ofselect=*"$partyId"*/></tcrmParam>

              </InquiryParam>

          </TCRMInquiry>

      </TCRMService>

   </xsl:template>

</xsl:stylesheet>

Sample xml template for retrieving the history data for Organization using the existing “getOrganization” inquiry(SearchOrganizationHistoryDetail.xsl), need to be called by SearchOrganizationHistoryDetail.getStylesheetName method:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!--

Stylesheet that creates a getPerson XML file that can be sent to the MDM Server.

 The information that the stylesheet needs is all passed in as parameters.

  -->

<xsl:stylesheet xmlns:xsl=*"http://www.w3.org/1999/XSL/Transform"* version=*"1.0"*>

   <xsl:param name=*"requestId"*/>

   <xsl:param name=*"requesterName"*/>

   <xsl:param name=*"requesterLanguage"*/>

    <xsl:param name=*"inquireAsOfDate"*/>

   <xsl:param name=*"id"*/>

   <xsl:output method=*"xml"*/>

   <xsl:template match=*"/"*>

      <TCRMService xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:noNamespaceSchemaLocation=*"KSCRequest.xsd"*>

          <RequestControl>

              <requestID><xsl:value-of select=*"$requestId"*/></requestID>

              <DWLControl>

                  <requesterName><xsl:value-of select=*"$requesterName"*/></requesterName>

                  <requesterLanguage><xsl:value-of select=*"$requesterLanguage"*/></requesterLanguage>

                  <inquireAsOfDate><xsl:value-of select=*"$inquireAsOfDate"*/></inquireAsOfDate>

              </DWLControl>

          </RequestControl>

          <TCRMInquiry>

              <InquiryType>getOrganization</InquiryType>

              <InquiryParam>

                 <tcrmParam name=*"PartyId"*><xsl:value-of select=*"$id"*/></tcrmParam>

                 <tcrmParam name=*"InquiryLevel"*>4</tcrmParam>

              </InquiryParam>

          </TCRMInquiry>

      </TCRMService>

   </xsl:template>

</xsl:stylesheet>

Sample xml for getting the End Date History value from Security History table(GetEndDateHistoryFromSecurityHistory.xsl), need to be called by GetEndDateHistoryFromSecurityHistory.getStylesheetName method:

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!--

Stylesheet that creates a getPerson XML file that can be sent to the MDM Server.

 The information that the stylesheet needs is all passed in as parameters.

  -->

<xsl:stylesheetxmlns:xsl=*"http://www.w3.org/1999/XSL/Transform"*version=*"1.0"*>

   <xsl:paramname=*"requestId"*/>

   <xsl:paramname=*"requesterName"*/>

   <xsl:paramname=*"requesterLanguage"*/>

   <xsl:paramname=*"securityId"*/>

   <xsl:outputmethod=*"xml"*/>

   <xsl:templatematch=*"/"*>

      <TCRMServicexmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xsi:noNamespaceSchemaLocation=*"KSCRequest.xsd"*>

          <RequestControl>

              <requestID><xsl:value-ofselect=*"$requestId"*/></requestID>

              <DWLControl>

                  <requesterName><xsl:value-ofselect=*"$requesterName"*/></requesterName>

                  <requesterLanguage><xsl:value-ofselect=*"$requesterLanguage"*/></requesterLanguage>

              </DWLControl>

          </RequestControl>

          <TCRMInquiry>

              <InquiryType>GetEndDateHistoryFromSecurityHistory</InquiryType>

              <InquiryParam>

                 <tcrmParamname=*"*securityId*"*><xsl:value-ofselect=*"$securityId"*/></tcrmParam>

              </InquiryParam>

          </TCRMInquiry>

      </TCRMService>

   </xsl:template>

</xsl:stylesheet>

Sample xml template for retrieving the history data for Security using the existing “getKSCSecurity” inquiry(SearchSecurityHistoryDetail.xsl), need to be called by SearchSecurityHistoryDetail.getStylesheetName method.

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!--

Stylesheet that creates a getPerson XML file that can be sent to the MDM Server.

 The information that the stylesheet needs is all passed in as parameters.

  -->

<xsl:stylesheetxmlns:xsl=*"http://www.w3.org/1999/XSL/Transform"*version=*"1.0"*>

   <xsl:paramname=*"requestId"*/>

   <xsl:paramname=*"requesterName"*/>

   <xsl:paramname=*"requesterLanguage"*/>

      <xsl:paramname=*"inquireAsOfDate"*/>

   <xsl:paramname=*"id"*/>

   <xsl:outputmethod=*"xml"*/>

   <xsl:templatematch=*"/"*>

      <TCRMServicexmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xsi:noNamespaceSchemaLocation=*"KSCRequest.xsd"*>

          <RequestControl>

              <requestID><xsl:value-ofselect=*"$requestId"*/></requestID>

              <DWLControl>

                  <requesterName><xsl:value-ofselect=*"$requesterName"*/></requesterName>

                  <requesterLanguage><xsl:value-ofselect=*"$requesterLanguage"*/></requesterLanguage>

                  <inquireAsOfDate><xsl:value-ofselect=*"$inquireAsOfDate"*/></inquireAsOfDate>

              </DWLControl>

          </RequestControl>

          <TCRMInquiry>

              <InquiryType>getKSCSecurity</InquiryType>

              <InquiryParam>

                 <tcrmParamname=*"id"*><xsl:value-ofselect=*"$id"*/></tcrmParam>

              </InquiryParam>

          </TCRMInquiry>

      </TCRMService>

   </xsl:template>

</xsl:stylesheet>