Validation Rules for Corporate Actions

Cash Dividend Event

Required Fields
The following items must be tagged in the document:

- AnnouncementDate
- EventCompleteness
- UniqueUniversalEventIdentifier
- AnnouncementIdentifier
- AnnouncementType
- EventType
- InstrumentIdentifier
- RecordDate
- MandatoryVoluntaryChoiceIndicator

Dependent Fields

EventCompleteness
The following fields must be tagged if the EventCompleteness tag is tagged with a value of “Complete”.

- EventConfirmationStatus
- CountryOfIssuer
- PaymentDate
- OptionType

Consistent Values
The following fields must have an appropriate value for a Cash Dividend:

- OptionType must equal “Cash”.
- PayoutType must equal “Dividend”
- PayoutAmount >= PayoutAmountNetOfTax
- PayoutAmount must be greater than 0.
- PayoutAmountNetOfTax must be greater than 0
- Payment Date must be after the RecordDate
- OrdPaymentDate must be after the OrdRecordDate
- EventType must be equal to “Cash Dividend” or “Sale Of Rights”
If Payment Date is used at the event level and the option level the value should be the same.

**Inconsistent Meaning with the event**

For a mandatory cash dividend only one cash option can be defined as there is only one option and only one tax reclaim method is used. Multiple options can be used in the event if multiple tax rates are defined.

For each option there must be at least one payout if the Event Completeness indicator is set to “Complete”. 
Stock Dividend Event

Required Fields
The following items must be tagged in the document:

- AnnouncementDate
- EventCompleteness
- UniqueUniversalEventIdentifier
- AnnouncementIdentifier
- AnnouncementType
- EventType
- InstrumentIdentifier
- RecordDate
- MandatoryVoluntaryChoiceIndicator
- DisbursedQuantity
- BaseQuantity

Dependent Fields

EventCompleteness
The following fields must be tagged if the EventCompleteness tag is tagged with a value of “Complete”.

- EventConfirmationStatus
- CountryOfIssuer
- PaymentDate
- OptionType
- InstrumentIdentifier for the payout security.

Consistent Values
The following fields must have an appropriate value for a Cash Dividend:

- OptionType must equal “Securities”.
- PayoutType must equal “Dividend”
- PayoutAmount >= PayoutAmountNetOfTax
- PayoutAmount must be greater than 0.
- PayoutAmountNetOfTax must be greater than 0
- Payment Date must be after the RecordDate
- OrdPaymentDate must be after the OrdRecordDate
- EventType must be equal to “Stock Dividend”
- If Payment Date is used at the event level and the option level the value should be the same.
**Inconsistent Meaning with the event**

For a mandatory stock dividend only one option can be defined as there is only one option and only one tax reclaim method is used. Multiple options can be used in the event if multiple tax rates are defined.

For each option there must be at least one payout if the Event Completeness indicator is set to “Complete”. 
Cancellation of an Event

Required Fields
The following items must be tagged in the document:

- AnnouncementDate
- EventCompleteness
- UniqueUniversalEventIdentifier
- AnnouncementIdentifier
- AnnouncementType
- EventType
- MandatoryVoluntaryChoiceIndicator

Dependent Fields

EventCompleteness
The following fields must be tagged if the EventCompleteness tag is tagged with a value of “Complete”.

- EventConfirmationStatus
- CountryOfIssuer

Consistent Values
The following fields must have an appropriate value for a Cash Dividend:

- EventType must be equal to the name of the event being cancelled
- EventCompleteness must be equal to “Complete”
- EventConfirmationStatus must equal “Confirmed”
- All values in the message must use the ConfirmedMember.
raise xbrlus-cc.ca.PreliminaryAndComplete.100 severity error
   exists([StatusAxis=PreliminaryMember]#fact)
   and
   any(values EventCompleteness[] == 'Complete')
message
   "The element ${fact.context} with a value of ${fact.value} has a preliminary status, but the event is indicated to be complete. Please revise the status to confirmed or change the event completeness indicator to be incomplete. (id:100)\n   \n   \n
raise xbrlus-cc.ca.PreliminaryAndComplete.101 severity error
   exists([StatusAxis=UnconfirmedMember]#fact)
   and
   any(values EventCompleteness[] == 'Complete')
message
   "The element ${fact.context} with a value of ${fact.value} has an unconfirmed status, but the event is indicated to be complete. Please revise the status to confirmed or change the event completeness indicator to be incomplete. (id:100)\n   \n   \n
Cash Dividend

-- Corporate Action  Rules
-- copyright (c) 2012
--
-- Taxonomy: Corporate Actions 2012
-- Version: 2012-03-31

-- Define Constants
constant EventTypeMap = { {CashDividendMember,"Cash Dividend"},
    (CashDividendMember, "Sale Of Rights"),
    (StockDividendMember, "Stock Dividend"),
    (SpecialDividendMember, "Special Dividend"),
    (CashDividendWithCurrencyOptionMember, "Cash Dividend with Currency Option"),
    (DividendWithOptionMember, "Dividend with Option"),
    (CancelMember, "Sale Of Rights"),
    (CancelMember, "Annual General Meeting"),
    (CancelMember, "Assimilation"),
    (CancelMember, "Attachment"),
    (CancelMember, "Automatic Dividend Reinvestment"),
    (CancelMember, "Bankruptcy Note"),
    (CancelMember, "Bankruptcy Vote"),
    (CancelMember, "Bankruptcy"),
    (CancelMember, "Bearer to Registered Form"),
    (CancelMember, "Bid Tender / Sealed Tender"),
    (CancelMember, "Bonus Issue"),
    (CancelMember, "Bonus Rights Issue"),
    (CancelMember, "Buy Up"),
    (CancelMember, "Capital Distribution"),
    (CancelMember, "Capital Gains Distribution"),
    (CancelMember, "Capitalisation"),
    (CancelMember, "Cash and Securities Merger"),
    (CancelMember, "Cash Dividend with Currency Option"),
    (CancelMember, "Cash Dividend"),
    (CancelMember, "Cash in Lieu"),
    (CancelMember, "Cash Merger"),
    (CancelMember, "Change in Board Lot"),
    (CancelMember, "Change in Domicile"),
    (CancelMember, "Change in Name"),
    (CancelMember, "Change in Place of Incorporation"),
    (CancelMember, "Change in Place of Listing"),
    (CancelMember, "Change In Security Term"),
    (CancelMember, "Change Resulting in Decrease of Par Value"),
    (CancelMember, "Change Resulting in Increase of Par Value"),
    (CancelMember, "Class Action"),
    (CancelMember, "Consent for Plan of Reorganization"),
    (CancelMember, "Consent Tender"),
    (CancelMember, "Consent with No Payout"),
    (CancelMember, "Consent with Payout"),
    (CancelMember, "Convert And Tender"),
    (CancelMember, "Convertible Security Issue"),
    (CancelMember, "Coupon Distribution"),
    (CancelMember, "Credit Event"),
    (CancelMember, "Decimalisation"),
    (CancelMember, "Default"),
    (CancelMember, "Dematerialised to Physical Form"),
    (CancelMember, "Dissent"),
(CancelMember, "Distribution on Recapitalization"),
(CancelMember, "Distribution"),
(CancelMember, "Dividend Reinvestment"),
(CancelMember, "Dividend with Option"),
(CancelMember, "Drawing"),
(CancelMember, "Dutch Auction Tender"),
(CancelMember, "Dutch Auction"),
(CancelMember, "Exchange Offer with Consent Fee"),
(CancelMember, "Exchange Offer"),
(CancelMember, "Exchange on 144a Type Securities"),
(CancelMember, "Exchange on Reg S Type Securities"),
(CancelMember, "Exercise"),
(CancelMember, "Extraordinary General Meeting"),
(CancelMember, "Final Paydown"),
(CancelMember, "Full Call on Convertible Security"),
(CancelMember, "Full Call"),
(CancelMember, "Full Pre-refunding"),
(CancelMember, "General Information"),
(CancelMember, "Global Permanent to Physical Form"),
(CancelMember, "Global Temporary to Global Permanent Form"),
(CancelMember, "Global Temporary to Physical Form"),
(CancelMember, "Holdings Disclosure"),
(CancelMember, "Interest"),
(CancelMember, "Issue Fraction"),
(CancelMember, "Liquidation"),
(CancelMember, "Mandatory (Put) Tender"),
(CancelMember, "Mandatory (Put) With Option to Retain"),
(CancelMember, "Mandatory Exchange"),
(CancelMember, "Mandatory Redemption of Shares"),
(CancelMember, "Mandatory Tender"),
(CancelMember, "Maturity Extension"),
(CancelMember, "Maturity"),
(CancelMember, "Meeting"),
(CancelMember, "Merger"),
(CancelMember, "Mini Tender"),
(CancelMember, "Mortgage Backed"),
(CancelMember, "Non US TEFRAD Certification"),
(CancelMember, "Odd Lot Offer"),
(CancelMember, "Offer To Purchase"),
(CancelMember, "Ordinary Meeting"),
(CancelMember, "Par Value Change"),
(CancelMember, "Partial Call on Convertible Security"),
(CancelMember, "Partial Call With Reduction in Nominal Value"),
(CancelMember, "Partial Call"),
(CancelMember, "Partial Defeasance"),
(CancelMember, "Partial Mandatory (Put) Tender"),
(CancelMember, "Partial Mandatory Tender"),
(CancelMember, "Partial Prerefunding"),
(CancelMember, "Pay in Kind"),
(CancelMember, "Physical to Dematerialised Form"),
(CancelMember, "Physical to Dematerialized Form"),
(CancelMember, "Principal"),
(CancelMember, "Put"),
(CancelMember, "Redemption"),
(CancelMember, "Redenomination"),
(CancelMember, "Registered to Bearer Form"),
(CancelMember, "Remarketing Agreement"),
(CancelMember, "Remarketing"),
(CancelMember, "Reorganization"),
(CancelMember, "Return of Capital"),
// Pre Condition
precondition us-Equity-CashDiv
let firstfact = first(values[])
in
$firstfact::dimension(EventTypeAxis) == CashDividendMember and $firstfact::dimension(MarketTypeAxis) == UnitedStatesMember and $firstfact::dimension(IssueTypeAxis) == EquityMember and $firstfact::dimension(MandatoryVoluntaryAxis) == MandatoryMember

raise xbrlus-cc.ca.exists.1 severity error
count(values[primary=AnnouncementDate]) == 0
message "Value element AnnouncementDate must exist in the document.\n(id:1)"
raise xbrlus-cc.ca.exists.2 severity error
count(values[primary= EventCompleteness]) == 0
message "The element EventCompleteness must exist in the document.\n(id:2)"
raise xbrlus-cc.ca.exists.3 severity error
count(values[primary= UniqueUniversalEventIdentifier]) == 0
message "the element UniqueUniversalEventIdentifier must exist in the document.\n(id:3)"
raise xbrlus-cc.ca.exists.4 severity error
count(values[primary=AnnouncementIdentifier]) == 0
message
"The element AnnouncementIdentifier must exist in the document.\n(id:4)"
raise xbrlus-cc.ca.exists.5 severity error
count(values[primary=AnnouncementType]) == 0
message
"The element AnnouncementType must exist in the document.\n(id:5)"
raise xbrlus-cc.ca.exists.6 severity error
count(values[primary=EventType]) == 0
message
"The element EventType must exist in the document.\n(id:6)"
raise xbrlus-cc.ca.exists.9 severity error
count(values[primary=MandatoryVoluntaryChoiceIndicator]) == 0
message
"The element MandatoryVoluntaryChoiceIndicator must exist in the document.\n(id:9)"
require us-Equity-CashDiv
raise xbrlus-cc.ca.exists.7 severity error
count(values[primary=InstrumentIdentifier;UnderlyingSecuritiesImpactedTypedAxis="",UnderlyingInstrumentIdentifierSchemeAxis=""]
) == 0
message
"The element InstrumentIdentifier must exist in the document for the security impacted by the corporate action.\n(id:7)"
require us-Equity-CashDiv
raise xbrlus-cc.ca.exists.8 severity error
count(values[primary=RecordDate]) == 0
message
"The element RecordDate must exist in the document.\n(id:8)"
raise xbrlus-cc.ca.completedExists.10 severity error
[primary=EventCompleteness]#fact == "Complete"
and
count(values[primary=EventConfirmationStatus]) == 0
message
"The Event Confirmation Status must be populated in the document if the document is complete.\n(id:10)"
raise xbrlus-cc.ca.duplicateValue.11 severity error
count([#fact]>1
message
"The element $\{fact.context\} appears multiple times in the document.\n(id:11)"
raise us-Equity-CashDiv
require us-Equity-CashDiv
raise xbrlus-cc.ca.completedExists.12 severity error
[primary=EventCompleteness]#fact == "Complete"
and
missing([primary=CountryOfIssuer])
message
"The Country Of Issuer must be populated in the document for a cash dividend if the document is complete.\n(id:12)"
raise us-Equity-CashDiv
require us-Equity-CashDiv
raise xbrlus-cc.ca.completedExists.14 severity error
[primary=EventCompleteness]#fact == "Complete"
and
missing([primary=PaymentDate])
message
"The Payment Date must be populated in the document if the document is complete.\n(id:14)"
require us-Equity-CashDiv
raise xbrlus-cc.ca.completedExists.15 severity error
[primary=EventCompleteness]$fact == "Complete"
and
missing([primary=OptionType;EventOptionsSequenceTypedAxis=""])
message
"The Option Type must be populated in the document if the document is complete.\n(id:15)\n{${fact.context}\n${context}}"

require us-Equity-CashDiv
raise xbrlus-cc.ca.atLeastOneOptionIsRequired.16 severity error
count(values[primary=OptionType]$fact) < 1
message
"At least Option Type must be defined for a mandatory cash dividend.\n(id:16)\n{${fact.context}\n${context}}"

require us-Equity-CashDiv
raise xbrlus-cc.ca.noMoreThanOneOption.16a severity error
count(values[primary=OptionType]$fact) != count(values[primary = TaxRateDescription])
and
count(values[primary=OptionType]$fact) > 1
and
set(values WithholdingTaxPercentage[]):size#fact2 != count(values[primary=OptionType]$fact3)
message
"More than one option has been defined for a cash dividend only one Option can be defined for a mandatory cash dividend unless there is multiple tax rates defined. In the file there are $(fact2) unique withholding rates defined but $(fact3) options defined.\n(id:16a)\n{${fact.context}}"

raise xbrlus-cc.ca.paymentOptions.17 severity error
((primary=PayoutType)$fact) != "Dividend"
message
"The Payout Type for a cash dividend must be defined as a "Dividend".\n(id:17)\n{${fact.context}\n${context}}"

raise xbrlus-cc.ca.gte.18 severity error
((primary=PayoutAmount)$fact) < (primary=PayoutAmountNetOfTax $fact2)
message
"The $(fact.context) of $(fact.value) must always be greater than $(fact2.context) with a value of $(fact2.value).\n(id:18)\n{${fact.context}\n${context}}"

raise xbrlus-cc.us-equity-casDiv-mand.nonneg.19 severity error
((primary=PayoutAmount)$fact) < 0
message
"The $(fact.context) of $(fact.value) must always be positive.\n(id:19)\n{${fact.context}\n${context}}"

raise xbrlus-cc.ca.nonneg.21 severity error
((primary=PayoutAmountNetOfTax)$fact) < 0
message
"The $(fact.context) of $(fact.value) must always be positive.\n(id:21)\n{${fact.context}\n${context}}"

raise xbrlus-cc.us-equity-cashDiv-mand.date.22 severity error
instant([primary=PaymentDate]$fact) < instant([primary=RecordDate]$fact2)
message
"The $(fact.context) of $(fact.value) must always be a date later than the $(fact2.context) of $(fact2.value).\n(id:22)\n{${fact.context}\n${context}}"

raise xbrlus-cc.us-equity-cashDiv-mand.date.23 severity error
instant([primary=OrdPaymentDate]$fact) < instant([primary=OrdRecordDate]$fact2)
message
"The $(fact.context) of $(fact.value) must always be a date later than the $(fact2.context) of $(fact2.value).
(id:23)\n{$(fact.context)\n$(context)}"

require us-Equity-CashDiv
raise xbrlus-cc.us-equity-cashDiv-mand.eventTypeMatch.20 severity error
not $EventTypeMap::contains([primary=EventType].dimension(EventTypeAxis),[primary=EventType]#fact))
message
"Value $(fact.value) for $(fact.context) must be defined as 'Cash Dividend' or 'Sale Of Rights'. (id:20)\n{$(fact.context)\n$(context)}"

require us-Equity-CashDiv
raise xbrlus-cc.us-equity-cashDiv-mand.dupValues.24 severity error
[primary=PaymentDate]#fact2 != ([primary=PaymentDate;EventOptionsSequenceTypedAxis='1; PayoutSequenceTypedAxis = *']#fact)
message
"The element $(fact.context) with a value of $(fact.value) at the payout detail level must be equal to $(fact2.context) at the event level with a value of $(fact2.value). (id:24)\n{$(fact.context)\n$(context)}"

require us-Equity-CashDiv
raise xbrlus-cc.us-equity-cashDiv-mand.multiplePayouts.25 severity error
([primary=PaymentDate;EventOptionsSequenceTypedAxis='1; PayoutSequenceTypedAxis = *']#fact2) != ([primary=PaymentDate;EventOptionsSequenceTypedAxis='1; PayoutSequenceTypedAxis = *']#fact)
message
"The element $(fact.context) with a value of $(fact.value) at the payout detail level must be equal to $(fact2.context) at the payout level with a value of $(fact2.value). (id:25)\n{$(fact.context)\n$(context)}"

require us-Equity-CashDiv
raise xbrlus-cc.us-equity-cashDiv-mand.missingPayouts.25a severity error
count(values[primary=OptionType]#fact) > count(values [primary = PayoutType]) and EventCompleteness[] == 'Complete'
message
"The number of payouts associated with a cash dividend must match the number of options on a complete corporate action event. Each option must have at least one payout associated with it. (id:25a)\n{$(context)}"

require us-Equity-CashDiv
raise xbrlus-cc.ca.noMoreThanOnOption.16b severity error
count(values[primary=OptionType]#fact) > 1
and
max(values WithholdingTaxPercentage[][#fact2 != [primary=WithholdingTaxPercentage;EventOptionsSequenceTypedAxis='1']#fact3)
message
"In those cases where multiple tax rates are defined the highest rate typically represents the payout rate associated with the corporate action to the clearing and settlement organization. The first option should represent the distribution made for settlement. In this case the withholding tax rate for option 1 is $(fact3.value). This is not the maximum withholding rate associated with the action which is $(fact2.value). Make the payout with the highest withholding rate the first option. (id:16b)\n{$(context)}"

raise xbrlus-cc.ca.ne.26 severity error
([primary=PayoutAmount]#fact) < ([primary=PayoutAmountNetOfTax ]#fact2) + ([primary=TaxAmountWithheldFromPayout]#fact3)
message
"The $(fact.context) of $(fact.value) must always be greater than or equal to the sum of $(fact2.context) with a value of $(fact2.value) and $(fact3.context) with a value of $(fact3.value). (id:26)\n{$(fact.context)\n$(context)}"
Stock Dividend

// Corporate Action Rules
// copyright (c) 2012
//
// Taxonomy: Corporate Actions 2012
// Version: 2012-03-31

precondition us-Equity-StockDiv
let
    firstfact = first(values[])
in
    $firstfact::dimension(EventTypeAxis)== StockDividendMember
    and $firstfact::dimension(MarketTypeAxis)== UnitedStatesMember
    and $firstfact::dimension(IssueTypeAxis)== EquityMember
    and $firstfact::dimension(MandatoryVoluntaryAxis)== MandatoryMember

//-----------------STOCK DIVIDEND SPECIFIC ----------------
require us-Equity-StockDiv
raise xbrlus-cc.us-equity-stockDiv.onlyOneOptionAllowed.41 severity error
    count(values[primary=OptionType]#fact) != 1
message
"Only one Option Type can be defined for a mandatory stock dividend.
\n(id:41)\n$\{fact.context\}\n$\{context\}\""

require us-Equity-StockDiv
raise xbrlus-cc.ca.noMoreThanOnOption.41a severity error
    count(values[primary=OptionType]#fact) != count(values [primary = TaxRateDescription])
    and
    count(values[primary=OptionType]#fact)>1
    and
    set(values WithholdingTaxPercentage[])::size != count(values[primary=OptionType])
message
"More than one option has been defined for a stock dividend. Only one Option can be defined for a mandatory stock dividend unless there is multiple tax rates defined.
\n(id:41a)\n$\{fact.context\}\n$\{context\}\"

require us-Equity-StockDiv
raise xbrlus-cc.ca.completedExists.50 severity error
    [primary=EventCompleteness]#fact == "Complete"
    and
    missing([primary=CountryOfIssuer])
message
"The Country Of Issuer must be populated in the document for a stock Dividend if the document is complete.
\n(id:50)\n$\{fact.context\}\n$\{context\}\"

require us-Equity-StockDiv
raise xbrlus-cc.ca.completedExists.48 severity error
    [primary=EventCompleteness]#fact == "Complete"
    and
    missing([primary=OptionType;EventOptionsSequenceTypedAxis=*])
message
"The Option Type must be populated in the document if the document is complete.
\n(id:48)\n$\{fact.context\}\n$\{context\}\"

require us-Equity-StockDiv
raise xbrlus-cc.ca.exists.40 severity error
    count(values[primary=InstrumentIdentifier;UnderlyingSecuritiesImpactedTypedAxis=*,UnderlyingInstrumentIdentifierSchemeAxis=]*)
    ) == 0
message
"The element InstrumentIdentifier must exist in the document for the security impacted by the corporate action.
\n(id:40)\n$\{"context\}\"

require us-Equity-StockDiv
raise xbrlus-cc.us-equity-stockDiv.exists.42 severity error
    count(values[primary=RecordDate]) == 0
message
"The element RecordDate must exist in the document.
\n(id:42)\n$\{context\}\"

require us-Equity-StockDiv
raise xbrlus-cc.ca.completedExists.43 severity error
[primary=EventCompleteness]#fact == "Complete"
and
missing([primary=PaymentDate])
message
"The Payment Date must be populated in the document if the document is complete.\n(id:43)\n{\$fact.context}"

require us-Equity-StockDiv
raise xbrlus-cc.us-equity-stockDiv.paymentOptions.44 severity error
[(primary=PayoutType)#fact] != "Dividend"
message
"The Payout Type for a cash dividend must be defined as a "Dividend".\n(id:44)\n{\$fact.context}"

require us-Equity-StockDiv
raise xbrlus-cc.us-equity-stockDiv.optionTypeValue.45 severity error
[(primary=OptionType)#fact] != "Securities"
message
"The Option Type for a mandatory stock dividend must be defined as "Securities".\n(id:45)\n{\$fact.context}"

require us-Equity-StockDiv
raise xbrlus-cc.us-equity-stockDiv-mand.eventTypeMatch.46 severity error
not $EventTypeMap::contains({[primary=EventType]::dimension(EventTypeAxis),[primary=EventType]#fact})
message
"Value $\{fact.value\} for $\{fact.context\} must be defined as "Stock Dividend".(id:46)\n{\$fact.context}"

require us-Equity-StockDiv
raise xbrlus-cc.us-equity-stockDiv-mand.dupValues.47 severity error
[primary=PaymentDate]#fact2 != ([primary=PaymentDate;EventOptionsSequenceTypedAxis=*; PayoutSequenceTypedAxis = *] #fact)
message
"The element $\{fact.context\} with a value of $\{fact.value\} at the payout detail level must be equal to $\{fact2.context\} at the event level with a value of $\{fact2.value\} .\n(id:47)\n{\$fact.context}"

require us-Equity-StockDiv
raise xbrlus-cc.ca.completedExists.49 severity error
count(values[primary=InstrumentIdentifier;PayoutSequenceTypedAxis=*;PayoutSecurityIdentifierSchemeAxis=*]) == 0
and
[primary=EventCompleteness]#fact == "Complete"
message
"The element InstrumentIdentifier must exist in the document for the security paid out as part of the corporate action if the event is complete.\n(id:49)\n"

require us-Equity-StockDiv
raise xbrlus-cc.us-equity-stockDiv-mand.missingPayouts.49a severity error
count(values[primary=OptionType]#fact) > count(values [primary = PayoutType]) and EventCompleteness[] == 'Complete'
message
"The number of payouts associated with a stock dividend must match the number of options on a complete corporate action event. Each option must have at least one payout associated with it.\n(id:49a)\n{\$context}"

require us-Equity-StockDiv
raise xbrlus-cc.us-equity-stockDiv-mand.missingPayouts.49b severity error
count(values[primary=DisbursedQuantity;PayoutSequenceTypedAxis=*.]) == 0
and
[primary=EventCompleteness]#fact == "Complete"
message
"The element DisbursedQuantity must exist in the document for the security paid out as part of the corporate action if the event is complete.\n(id:49b)\n"

require us-Equity-StockDiv
raise xbrlus-cc.us-equity-stockDiv-mand.missingPayouts.49c severity error
count(values[primary=BaseQuantity;PayoutSequenceTypedAxis=*.]) == 0
and
[primary=EventCompleteness]#fact == "Complete"
message
"The element BaseQuantity must exist in the document for the security paid out as part of the corporate action if the event is complete.\n(id:49c)\n"
// Define Constants

// Pre Condition
precondition cancel
let
  firstfact = first(values[])
in
    $firstfact::dimension(EventTypeAxis)== CancelMember
    and $firstfact::dimension(MarketTypeAxis)== CancelMember
    and $firstfact::dimension(IssueTypeAxis)== CancelMember
    and $firstfact::dimension(MandatoryVoluntaryAxis)== CancelMember

require cancel
raise xbrlus-cc.ca.cancelComplete.30 severity error

[primary=EventCompleteness]$fact != "Complete"

message
  "The Details Completness Status (EventCompleteness) must have a value of Complete for a cancel event.\nThe context is
${context}\n(id:30)\n{${fact.context}\n${context}}"

require cancel
raise xbrlus-cc.ca.exists.31 severity error

count(values[primary=EventCompleteness]) == 0

message
  "The Details Completness Status must must be tagged in the cancel document with a value of Complete.\nThe context is
${context}\n(id:31)"

require cancel
raise xbrlus-cc.ca.cancelComplete.32 severity error

[primary=EventConfirmationStatus]$fact != "Confirmed"

message
  "The Event Confirmation Status must be tagged with a value of Confirmed for a cancel event.\nThe context is
${context}\n(id:32)\n{${fact.context}\n${context}}"

require cancel
raise xbrlus-cc.she.invalid_member.33 severity error

exists([StatusAxis=UnconfirmedMember]$fact)

message
  "${fact.context} with a value of ${fact.value} has been reported with the UnconfirmedMember on the StatusAxis for a
cancel event.\nA cancel event must use the ConfirmedMember on the StatusAxis\n(id:33)\n{${fact.context}\n$context}"

require cancel
raise xbrlus-cc.she.invalid_member.34 severity error
exists([StatusAxis=PreliminaryMember; EventTypeAxis=CancelMember]#fact)

message "${fact.context} with a value of ${fact.value} has been reported with either the PreliminaryMember on the StatusAxis for a cancel event. A cancel event must use the ConfirmedMember on the StatusAxis\n(id:34)\n${fact.context}\n${context})"*

require cancel
raise xbrlus-cc.she.invalid_member.35 severity error

exists([StatusAxis=PreliminaryMember; EventTypeAxis=CancelMember]#fact)

message "${fact.context} with a value of ${fact.value} has been reported with the PreliminaryMember on the StatusAxis for a cancel event. A cancel event must use the ConfirmedMember on the StatusAxis\n(id:35)\n${fact.context}\n${context})"*

require cancel
raise xbrlus-cc.she.invalid_member.36 severity error

exists([StatusAxis=UnconfirmedMember; EventTypeAxis=CancelMember]#fact)

message "${fact.context} with a value of ${fact.value} has been reported with the UnconfirmedMember on the StatusAxis for a cancel event. A cancel event must use the ConfirmedMember on the StatusAxis\n(id:36)\n${fact.context}\n${context})"*

require cancel
raise xbrlus-cc.us-cancel.eventTypeMatch.37 severity error
not $EventTypeMap::contains([primary=EventType]:dimension(EventTypeAxis),[primary=EventType]#fact))

message "Value ${fact.value} for ${fact.context} must be defined as \"Stock Dividend\".(id:37)\n${fact.context}\n${context})"*