|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | **Version 1.0**  **CFS LCX Requirements** | | |
| **Sep 5, 2012** | |
|  | |
| **ID** | **Summary** |
|  | |
|  | |
| 19601 | CFS LCX Requirements Document |
| SR\_Contains:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ID** | **ReqID** | **Text** | **Rationale** | **Heritage\_Reference** | | 19603 |  |  |  |  | | 20927 |  |  |  |  | | 20928 |  | CFS Limit Checker eXtended (LCX) Requirements1.0       Introduction1.1        Document Purpose The Core Flight Software System (CFS) Limit Checker eXtended (LCX) Application is an extension from the original CFS Limit Checker (LC) Application. The LCX extension will be developed by the Flight Software Branch (FSB) of the Software Engineering Division (SED).  The purpose of this requirements specification is to define the requirements to be satisfied by the Limit Checker eXtended Application.  This application is developed for re-use.  For this reason, several nomenclatures are used in this document to identify configurations for a mission.  The CFS is specified as a multi-platform product. Mission-specific features and customization requirements which are applicable for all platforms are tagged with <MISSION\_DEFINED>.  Platform-specific features and customizations requirements are tagged with either “<PLATFORM\_DEFINED>” or “<OPTIONAL>.”  Additional nomenclature is used along with the tag to specify a CFS default value for the platform-specific feature: “<PLATFORM\_DEFINED, Default\_Value>”.  Reference platforms (single processor and multi-processor architectures) are defined to supply the default CFS application configuration.  These configurations define the “maximum” CFS Application deployments such that any refined deployment is a subset of a reference platform. 1.2        Document Scope The scope of this document is limited to the specification of requirements for the Limit Checker eXtended Software.  These include functional, performance*,* qualification, and design requirements. 1.3        Document Organization This document is organized into three additional sections and several appendices.  Section 2 gives the Limit Checker eXtended context.  Section 3 documents the Limit Checker eXtended system design decisions and constraints.  Section 4 contains the Limit Checker eXtended functional and performance requirements.  Appendix A contains a list of abbreviations and acronyms used in this document. 1.4        Relevant Documents1.4.1     Parent Documents CFS Limit Checker Application Heritage Analysis    582-2008-007 1.4.2     Reference Documents 1.        Operating System Abstraction Layer (OSAL) Library  2.        cFE Application Developer’s Guide   582-2007-001  3.        cFE User’s Guide 2.0       CFS Limit Checker eXtended Application Context In the most basic sense, the Limit Checker eXtended application monitors telemetry data points in the flight system and compares the values against predefined or computed threshold limits. When a threshold condition is encountered, an event message is issued and a Relative Time Sequence (RTS) command script may be initiated to respond to the threshold violation.  The definitions of the data to monitor and the actions to be taken are defined in 2 separate tables (see Figure 1.0):  ·         Watchpoint Definition Table  (WDT)– defines the data to be evaluated  (message id, offset, comparison value etc)  ·         Actionpoint Definition Table (ADT) – defines the equations to use to evaluate the data and the actions to be taken  Figure 1.0 shows the context diagram for the CFS Limit Checker eXtended (LCX) Application.  During initialization, LCX subscribes to messages from other applications as defined in the WDT. The Scheduler Application (SCH) sends periodic commands to LCX as defined in the SCH Schedule Table. Ground commands come from the Command Ingest task (CI). Messages are routed to LCX by the cFE SB Application. LC learns of ground updates to the LC tables through the cFE Table Services application. Actions taken by LC as defined in the ADT are sent to the CFS Stored Command  (SC) Application.  Messages generated by LCX are routed to Housekeeping (HK) and Data Store (DS) (as long as the applications subscribe to them).  LCX generates 2 dump-only tables:  ·         Watchpoint Results Table  (WRT)– Contains the results of the Watchpoint evaluations defined in the WDT  ·         Actionpoint Results Table (ART) – Contains the results of the Actionpoint evaluations defined in the ADT    **Figure 1.0 – CFS LCX Context** 2.1        Assumptions The following list summarizes the assumptions made by the CFS Limit Checker eXtended Application:  ·         cFE API and OSAL are  being used  ·         A command is sent to LCX to schedule the execution of the application (from Scheduler Application) 3.0       Design Specifications The Limit Checker eXtended Application’s requirements and design are identical to the CFS Limit Checker Application with one enhancement. LCX adds support for "stale" Watchpoint and Actionpoint results. The Limit Checker Application's requirements and design are based on the results of the CFS heritage analysis effort. The results of the heritage analysis are documented in the CFS Limit Checker Application Heritage Analysis document.  ·         Receipt of messages.  LCX subscribes to the messages defined in the WDT.  Upon receipt of each message, LCX processes the messages based on the definitions in the WDT and will reset the age of the Watchpoint back to the limit specified in the WDT.  ·         Receipt of an LCX Sample command.  Typically this command is contained in the Scheduler Application table.  When LCX receives this command, if the Watchpoint Age Update argument is set to TRUE, LCX will evaluate the actionpoints as defined in the ADT and decrement the Watchpoint Stale Results Counters. When a Watchpoint Stale Results Counter decrements to zero, LCX will update the Watchpoint result for that Watchpoint to be "Stale".    Note that custom functions can be defined if the operators (<,  <=,  !=,  =,  >,  >=) do not provide enough functionality.    As Actionpoints are evaluated, their results and activities are stored real-time in this dump-only Table.    It is the responsibility of the operator to ensure that changes to the WDT and/or the ADT result in a valid configuration.     * **Active -** Active mode is the normal operation mode. In Active mode, the LCX application performs all limit tests defined in the WDT and will invoke stored command sequences if the results of Watchpoint evaluations trigger an Actionpoint condition as defined in the ADT.      * **Passive -** In Passive mode, the LCX application behaves just like in Active mode except no stored command sequences will be invoked as a result of Actionpoint triggers.      * **Disabled -** When disabled, the LCX application will perform no Watchpoint or Actionpoint limit tests.\     **In addition, LCX provides the capability to set the state of individual Actionpoints to Active, Passive or Disable.  Note that there is no capability to set the state of Watchpoints, only Actionpoints.** 3.1        Design Constraints Since the definition of the Watchpoints and Actionpoints is defined in two separate tables, careful attention needs to be paid to ensure that the tables are consistent with each other.   4.0       Subsystem Requirements |  |  | | 20932 | CFS-500 | The CFS shall provide for on-board monitoring of telemetry points. |  |  | | 20935 | CFS-501 | The CFS shall have the capability to modify the selection and configuration of  monitored telemetry via ground command |  |  | | 20936 | CFS-502 | The selection of the monitored points shall be table driven and configurable via ground uploads |  |  | | 20937 | CFS-503 | The CFS shall be able to execute autonomous recovery sequences upon detection of threshold failures in monitored telemetry |  |  | | 20938 | CFS-504 | The contents of recovery sequences shall be configurable via ground uploads |  |  | | 20939 |  | 5.0       Detailed Requirements  5.1        Basic Requirements The following requirements are basic requirements of Limit Checker eXtended Application.  Some of them are included here to avoid repeating these requirements for each applicable requirement. |  |  | | 19607 | LCX1000 | Upon receipt of a No-Op command, LCX shall increment the LCX Valid Command Counter and generate an event message. |  | LRO | | 19609 | LCX1001 | Upon receipt of a Reset command, LCX shall reset the following housekeeping variables to a value of zero:  a) Valid Command Counter  b) Command Rejected Counter  c) Passive RTS Execution Counter  d) Actionpoint Sample Count  e) TLM Count  f) RTS Execution Counter |  | LRO, SDO | | 19611 | LCX1002 | For all LCX commands, if the length contained in the message header is not equal to the expected length, LCX shall reject the command and issue an event message. |  | LRO, SDO | | 19613 | LCX1003 | If LCX accepts any command as valid, LCX shall execute the command, increment the LCX Valid Command Counter and issue an event message |  | LRO, SDO | | 19615 | LCX1004 | If LCX rejects any command, LCX shall abort the command execution, increment the LCX Command Rejected Counter and issue an error event message |  | LRO, SDO | | 19617 | LCX2000 | The flight software shall monitor a maximum of <PLATFORM\_DEFINED> Watchpoints. |  | LRO, SDO | | 19619 | LCX2001 | For each Watchpoint specified in the Watchpoint Definition Table (WDT) LCX shall specify an age value which indicates when the data becomes “stale”. |  |  | | 19621 | LCX2002 | For each Watchpoint specified in the Watchpoint Definition Table (WDT) LCX shall maintain the age of the data. |  |  | | 19623 | LCX2003 | Upon receipt of a message, LCX shall compare the data in the message to the table-defined value using the table-defined comparison value and comparison operator for each data point defined in the Watchpoint Definition Table (WDT) if the LCX Application State is one of the following:  a)       Active  b)       Passive |  | LRO, (loosely SDO) | | 19625 | LCX2003.2 | LCX shall support the following comparison values:  a)       =  b)       !=  c)       >  d)       >=  e)       <  f)        <= |  |  | | 19627 | LCX2003.3 | If the WDT comparison operator specifies that a  Custom Function shall be performed, LCX shall apply the custom function to the data contained in the message |  | LRO, (loosely SDO) | | 19629 | LCX2003.4 | If the comparison result for a Watchpoint results in a False, LCX shall set the Number of Consecutive True values to zero |  | LRO, SDO | | 19631 | LCX2003.5 | If the Watchpoint cannot be evaluated, LCX shall set the Watchpoint Results Table to ERROR for the erroneous Watchpoint. |  | LRO | | 19633 | LCX2004 | For each Watchpoint, the flight software shall maintain the following statistics in the dump-only Watchpoint Results Table:  a)       The result of the last relational comparison (False, True, Error, or Stale)  b)       The number of times this Watchpoint has been compared  c)       The number of times this Watchpoint has crossed from the False to True result  d)       The number of consecutive times the comparison has yielded a True result  e)       The cumulative number of times the comparison has yielded a True result  f)        Most recent FALSE to TRUE transition value  g)       Most recent FALSE to TRUE transition timestamp  h)       Most recent TRUE to FALSE transition value  i)         Most recent TRUE to FALSE transition timestamp  j) Most recent comparison age |  | LRO, SDO | | 19635 | LCX2005 | Upon receipt of a table update indication, LCX shall validate the Watchpoint Definition Table for the following:  a)       Valid operator  b)       Data size  c)       Message  ID |  | LRO | | 19637 | LCX3000 | Limit Checker shall support up to a maximum of <PLATFORM\_DEFINED> Actionpoints |  |  | | 19639 | LCX3001 | Upon receipt of a Sample Request, LCX shall process the request specified Actionpoints defined in the Actionpoint Definition Table (ADT) if the LCX Application State is one of the following:  a)       Active  b)       Passive |  | LRO | | 19641 | LCX3001.1 | LCX shall support the following Reverse Polish Operators:  a)       and  b)       or  c)       xor  d)       not  e)       equals |  |  | | 19643 | LCX3001.2 | If the equation result for an Actionpoint results in a Pass, LCX shall set the Number of Consecutive Fail values to zero |  | LRO, SDO | | 19645 | LCX3001.3 | If the Actionpoint cannot be evaluated, LCX shall set the Actionpoint Results Table to ERROR for the erroneous Actionpoint. |  | LRO | | 19647 | LCX3002 | For Each table-defined Actionpoint, LCX shall store the results in the dump-only Actionpoint Results Table if the Actionpoint state is either:  a)       Active  b)       Passive |  | LRO, SDO (loosely) | | 19649 | LCX3002.1 | If the Actionpoint equation results in a transition from PASS to FAIL, LCX shall issue an event message indicating the failure |  | LRO, SDO | | 19651 | LCX3002.1.1 | If the PASS to FAIL transition event message has been sent for the table-defined number of times, LCX shall apply the table-defined event message filter. |  |  | | 19653 | LCX3002.2 | If the Actionpoint equation results in a transition from FAIL to PASS, LCX shall issue an event message indicating that the actionpoint is now within limits |  | LRO, SDO | | 19655 | LCX3002.2.1 | If the FAIL to PASS transition event message has been sent for the table-defined number of times, LCX shall apply the table-defined event message filter. |  |  | | 19657 | LCX3002.3 | If the equation has yielded a Fail result for the table-defined consecutive number of times limit and the Actionpoint is currently Active, LCX shall:  a)       generate an event message  b)       send a command to start the table-defined  RTS  c)       Increment the counter indicating Total count of commands sent to SC task to start an RTS |  | LRO, SDO | | 19659 | LCX3002.3.1 | Once an RTS is initiated, LCX shall change the current state of the associated Actionpoint to Passive. |  | LRO, SDO (loosely) | | 19661 | LCX3002.4 | If the equation has yielded a Fail result for the defined consecutive number of times and the Actionpoint is currently Passive, LCX shall  a)       generate an event message indicating that the Actionpoint Failed but the action was not taken  b)       Increment the counter indicating Number of Start RTS commands NOT sent to SC task because LCX Application is PASSIVE |  | LRO, SDO | | 19663 | LCX3003 | If the Actionpoint is Disabled, LCX shall skip processing that actionpoint |  | LRO, SDO (loosely) | | 19665 | LCX3004 | If the Actionpoint is Unused, LCX shall skip processing that actionpoint |  | LRO, SDO (loosely) | | 19667 | LCX3005 | If the Actionpoint is Permanently Disabled, LCX shall skip processing that actionpoint |  | LRO | | 19669 | LCX3006 | For each Actionpoint, the flight software shall maintain the following statistics in the dump-only Actionpoint Results Table:  a)       The result of the last Sample (Pass, Fail, Error, or Stale)  b)       The current state (PermOff, Disabled, Active, Passive, Unused)  c)       The number of times this Actionpoint has crossed from the Fail to Pass state  d)       The number of times this Actionpoint has crossed from the Pass to Fail state  e)       The number of consecutive times the equation result = Failed  f)        The cumulative number of times the equation result = Failed  g)       The cumulative count of the RTS executions  h)       Total number of event messages sent |  | LRO | | 19671 | LCX3007 | Upon receipt of a table update indication, LCX shall validate the Actionpoint Definition Table for the following:  a)       valid default state  b)       RTS number (in range)  c)       Event Type (DEBUG, INFO, ERROR, CRITICAL)  d)       Failure Count (in range)  e)       Action Equation syntax |  | LRO | | 19673 | LCX4000 | Upon receipt of a Set LCX Application State To Active Command, LCX shall set the state of the LCX Application to Active |  | LRO, SDO | | 19675 | LCX4001 | Upon receipt of a Set LCX Application State to Passive Command, LCX shall set the LCX Application State to Passive |  | LRO, SDO (SDO used the term Watch Mode) | | 19677 | LCX4002 | Upon receipt of a Set LCX Application State to Disable Command, LCX shall set the LCX Application State to Disabled |  | LRO, SDO | | 19679 | LCX4003 | Upon receipt of a Set Actionpoint to Active Command, LCX shall set the state for the command-specified Actionpoint to ACTIVE such that the actionpoint is evaluated and the table-defined actions are taken based on the evaluation |  | LRO, SDO | | 19681 | LCX4004 | Upon receipt of a Set All Actionpoints to Active Command, LCX shall set the state for all Actionpoints to ACTIVE such that the actionpoints are evaluated and the table-defined actions are taken based on the evaluation |  | LRO, SDO | | 19683 | LCX4005 | Upon receipt of a Set Actionpoint to Passive Command, LCX shall set the state for the command-specified Actionpoint to PASSIVE such that the actionpoint is evaluated, however, no actions are taken |  | LRO, SDO | | 19685 | LCX4006 | Upon receipt of a Set All Actionpoints to Passive Command, LCX shall set the state for the all Actionpoints to PASSIVE such that all actionpoints are evaluated, however, no actions are taken |  | LRO, SDO | | 19687 | LCX4007 | Upon receipt of a Set Actionpoint to Disabled Command, LCX shall set the state for the command-specified Actionpoint to DISABLED such that the actionpoints are not evaluated and no actions are taken |  | LRO, SDO | | 19689 | LCX4008 | Upon receipt of a Set All Actionpoints to Disabled Command, LCX shall set the state for all Actionpoint to DISABLED such that:  a)       the actionpoints are not evaluated  b)       no actions are taken  c)       no event messages generated. | Debug command to verify application is alive | LRO, SDO | | 19691 | LCX4009 | Upon receipt of a Set Actionpoint to Permanent Disable, LCX shall mark the command-specified Actionpoint such that the Actionpoint cannot be Activated | Important for testing and on-orbit flight operations in order to start with a “clean slate” | LRO | | 19693 | LCX4009.1 | If a command  is received to Activate an Actionpoint which has been permanently disabled, the command shall be rejected | Basic command verification in the event of  SEU or memory corruption | LRO | | 19695 | LCX4010 | Upon receipt of a Reset Actionpoint Statistics Command, LCX shall set to zero, all of the following Actionpoint Statistics for the command-specified Actionpoints :  a)       Total number of FAIL to PASS transitions  b)       Total number of PASS to FAIL transitions  c)       Number of consecutive FAIL results  d)       Total number of FAIL results  e)       Total number of RTS executions  f)        Total number of event messages sent  relating to that Actionpoint | Operators require feedback on command execution.  Note that this only applies to “ground commands” (i.e. does not include requests that come from the scheduler) | LRO, SDO | | 19697 | LCX4011 | Upon receipt of a Reset All Actionpoint Statistics Command, LCX shall set to zero, all of the following Actionpoint Statistics for all Actionpoints:  a)       Total number of FAIL to PASS transitions  b)       Total number of PASS to FAIL transitions  c)       Number of consecutive FAIL results  d)       Total number of FAIL results  e)       Total number of RTS executions  f)        Total number of event messages sent  relating to that Actionpoint | Operators require feedback on command execution |  | | 19699 | LCX4012 | Upon receipt of a Reset Watchpoint Statistics Command, LCX shall set to zero all of the following Watchpoint Statistics for the command-specified Watchpoints:  a)       Total sample count for this watchpoint  b)       Number of times result transitioned from FALSE to TRUE  c)       Number of consecutive TRUE results  d)       Total number of TRUE results  e)       Most recent FALSE to TRUE transition value  f)        Most recent FALSE to TRUE transition timestamp  g)       Most recent TRUE to FALSE transition value  h)       Most recent TRUE to FALSE transition timestamp | Need to define a limit for defining data structures | LRO, SDO | | 19701 | LCX4013 | Upon receipt of a Reset All Watchpoint Statistics Command, LCX shall set to zero, all of the following Watchpoint Statistics for all Watchpoints:  a)       Total sample count for this watchpoint  b)       Number of times result transitioned from FALSE to TRUE  c)       Number of consecutive TRUE results  d)       Total number of TRUE results  e)       Most recent FALSE to TRUE transition value  f)        Most recent FALSE to TRUE transition timestamp  g)       Most recent TRUE to FALSE transition value  h)       Most recent TRUE to FALSE transition timestamp | Stale data is datum for which a packet update has not been received and/or has not been received in a requisite interval specified by the age value.  In most cases, it indicates that a subsystem is turned off or some other condition has prevented packet generation.  The goal of tracking stale data by LCX is to prevent LCX from taking action on old or inaccurate information. |  | | 19703 | LCX8000 | LCX shall generate a housekeeping message containing the following:  a)       Valid  Command Counter  b)       Command Rejected Counter  c)       Number of Start RTS commands NOT sent to SC task because LCX Application is PASSIVE  d)       Current LCX Application State (LCX\_ACTIVE, LCX\_PASSIVE, LCX\_DISABLED)...  e)       Total count of actionpoints sampled while LCX\_ACTIVE or LCX\_PASSIVE...  f)        Total count of packets monitored for watchpoints (cmd and tlm)  g)       Total count of commands sent to SC task to start an RTS  h)       Selected data from watchpoint results table  i)         Selected data from actionpoint results table | Provides tracking of stale data. | LRO, SDO (loosely) | | 19705 | LCX9000 | Upon cFE Power-On LCX shall initialize the following Housekeeping data to Zero (or value specified):  a)       Valid Command Counter  b)       Command Rejected Counter  c)       Passive RTS Execution Counter  d)       Current LCX State to <PLATFORM\_DEFINED> Default Power-on State  e)       Actionpoint Sample Count  f)        TLM Count  g)       RTS Execution Counter  h)       Watch Results (bitmapped)  i)         Action Results (bitmapped) | Telemetry Watchpoints shall be evaluated whenever the message (packet) containing the data to be examined is received.  If LCX App is Disabled, LCX does not process WDT or ADT.   Disabled state useful for making updates to LCX tables (note that LCX does not have to be disabled in order to make WDT or ADT updates, however, care must be taken to ensure the validity of the relationship between the 2 tables during updates). Note that individual Watchpoints can’t be disabled;  Individual Actionpoints can be disabled.  Note also that WDT can have unused entries | Derived | | 19707 | LCX9001 | Upon cFE Power-On LCX shall initialize the following Watchpoint data to Zero (or value specified) for all Watchpoints:  a)       The result of the last watchpoint relational comparison to STALE  b)       The number of times this Watchpoint has been compared  c)       The number of times this Watchpoint has crossed from the False to True result  d)       The number of consecutive times the comparison has yielded a True result  e)       The cumulative number of times the comparison has yielded a True result  f)        The value that caused the last False-to-True crossing, and the crossing time stamp  g)       The value that caused the last True-to-False crossing, and the crossing time stamp | Watchdog comparison values |  | | 19709 | LCX9002 | Upon cFE Power-On LCX shall initialize the following Actionpoint data to Zero (or value specified for all Actionpoints:  a)       The result of the last Actionpoint Sample to STALE  b)       The current state as defined in the ADT  c)       The number of times this Actionpoint has crossed from the Fail to Pass state  d)       The number of times this Actionpoint has crossed from the Pass to Fail state  e)       The number of consecutive times the equation result = Failed  f)        The cumulative number of times the equation result = Failed  g)       The cumulative count of the RTS executions  h)       Total number of event messages sent | Provides capability to do a caLCXulation on the data instead of just a comparison |  | | 19711 | LCX9003 | Upon a cFE Processor Reset or LCX Application Reset,  if the <PLATFORM\_DEFINED> Save Critical Data parameter is set to NO, LCX shall perform the same initialization as a cFE Power-on (see LCX9000, LCX9001, LCX9002, and 9003) | Restarts the consecutive True values |  | | 19713 | LCX9004 | Upon a cFE Processor Reset or LCX Application Reset,  if the <PLATFORM\_DEFINED> Save Critical Data parameter is set to YES, LCX shall  restore the following data:  a)       LCX housekeeping data  b)       WDT  c)       Watchpoint Statistics  d)       ADT  e)       Actionpoint Statistics | Want to report that there was an erroneous watchpoint |  | | 19715 | LCX9004.1 | LCX shall initialize the LCX Application State to <PLATFORM\_DEFINED> Default Reset State | Provides results for each evaluation of the WDT. |  | | 19717 | LCX9004.1.1 | If the platform defined Default Reset State indicates to use the state of the LCX Application prior to the reset, LCX shall set the state of the LCX Application to the state restored from the CDS | Valid the items that can be validated when an table is updated (some items cannot be valued until runtime) |  | | 19719 | LCX9004.3 | If LCX determines the Critical Data is invalid, LCX shall perform the same initialization as a cFE Power-on (see LCX9000, LCX9001 and LCX9002) | Used to size the actionpoint table |  | | 19721 | LCX9005 | Upon any initialization, LCX shall validate the Watchpoint Definition Table for the following:  a)       valid operator  b)       data size  c)       Message ID | Processing of the Actionpoint Definition Table is scheduled upon receipt of a  command (usually from the CFS Scheduler Application).  If LCX App is Disabled, LCX does not process WDT or ADT.   Disabled state useful for making updates to LCX tables. Note that individual Watchpoints can’t be disabled;  Individual Actionpoints can be disabled.  Can sample individual or all actionpoints. | LRO | | 19723 | LCX9006 | Upon any initialization, LCX shall validate the Actionpoint Definition Table for the following:  a)       valid default state  b)       RTS number (in range)  c)       Event Type (DEBUG, INFO, ERROR, CRITICAL)  d)       Failure Count (in range)  e)       Action Equation syntax |  | LRO | | 19725 | LCX9007 | Upon any initialization, LCX shall subscribe to the messages defined in the WDT. | Used to track consecutive failures.  If consecutive failures > max then take action.  Need to reset if Actionpoint doesn’t fail. | LRO | | 19727 | LCX9007.1 | For a cFE Processor Reset, If the Save Critical Data parameter is YES, LCX shall subscribe to the messages defined in the WDT restored from the CDS | Potential problems with the Watchpoint or the equation. |  | | 19729 | LCX9007.2 | For an LCX Application Reset, If the Save Critical Data parameter is YES, LCX shall subscribe to the messages defined in the WDT restored from the CDS | Actionpoint is not evaluated if state is unused, disabled or permanently disabled |  | | | | | |
|  | |  |  |  |