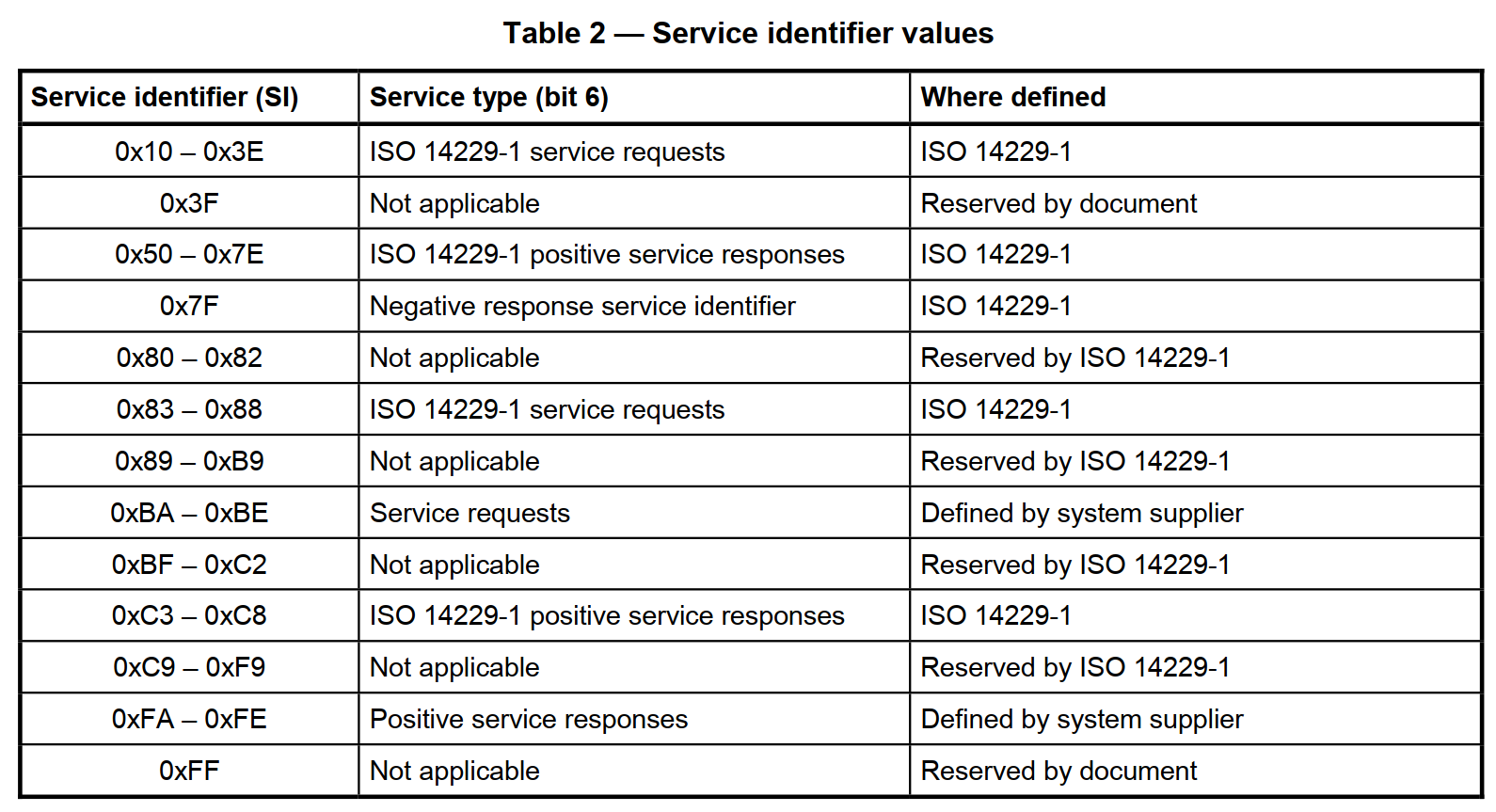
# 14229-1 2013 总结

7.3.2 SI, 服务ID



注意: 一个服务请求对应一个积极服务响应。SI的bit6代表着服务类型。所有的请求消息bit6=0，所有的积极响应bit6=1（除了周期性数据响应消息—0x2A 10.5）

举例： 7E0 02 10 01 CC CC CC CC CC

7E8 02 50 01 CC CC CC CC CC

7.3.3 消极响应

类型: 1个字节整型数据

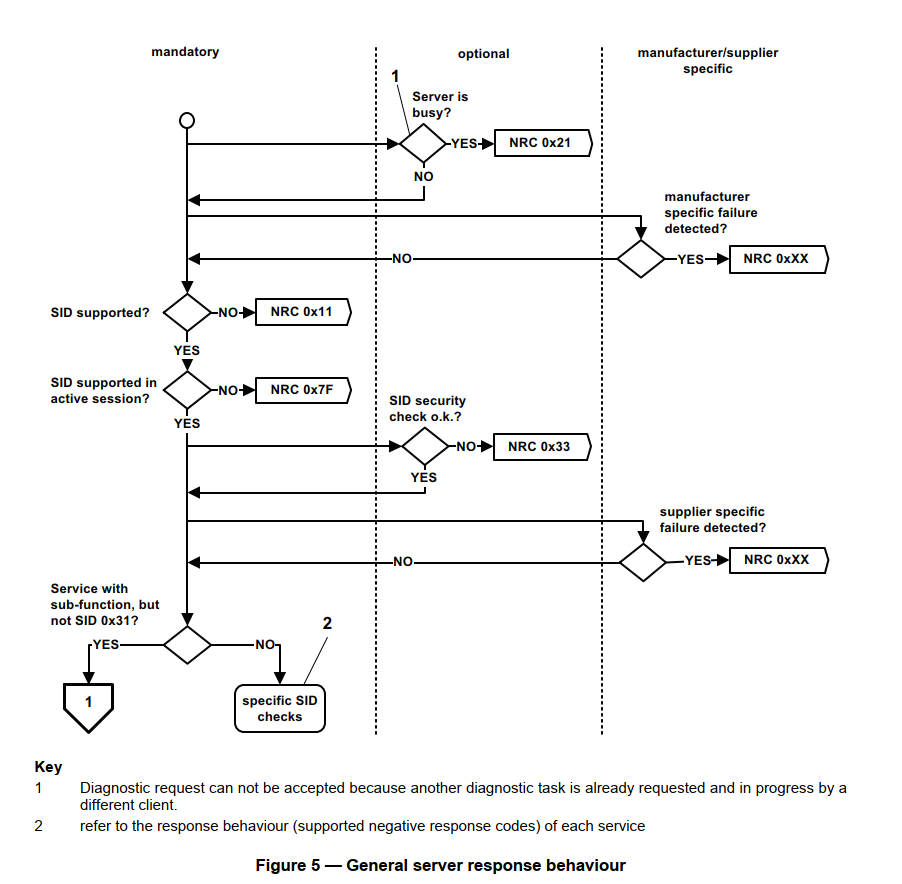
固定值: 0x7F

举个例子:

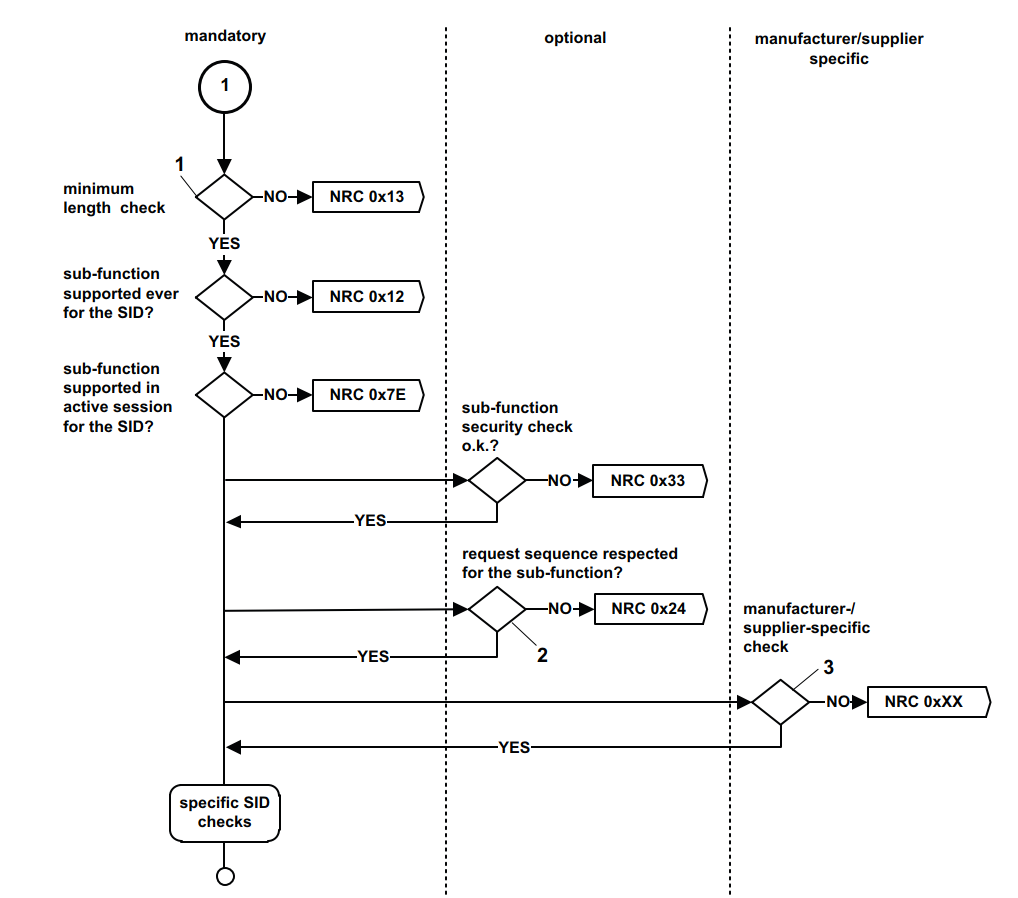
举例： 7E0 02 10 01 CC CC CC CC CC

7E8 03 7F 10 11 CC CC CC CC

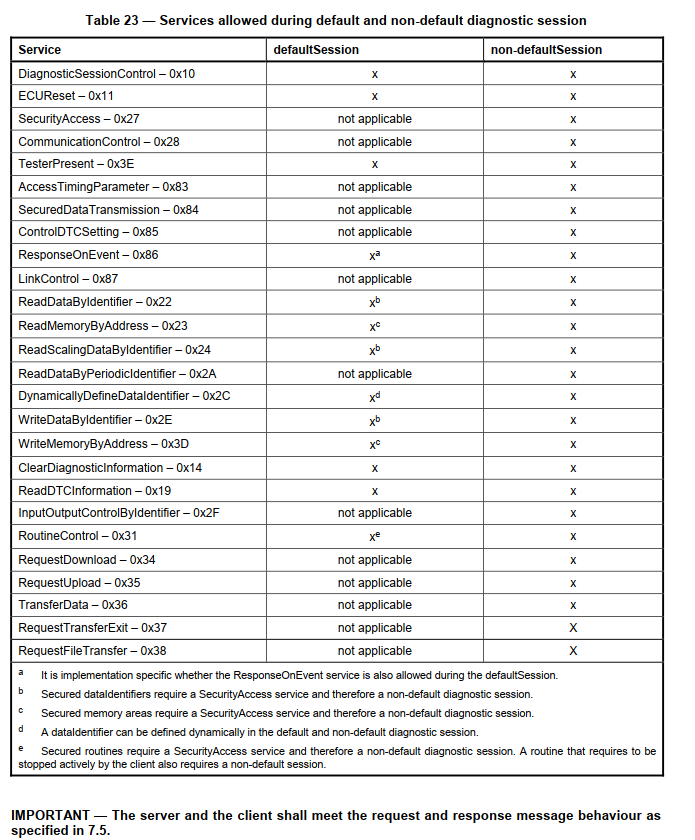
消极响应的操作流程



7.5.3 带有子功能参数的请求消息和服务响应行为

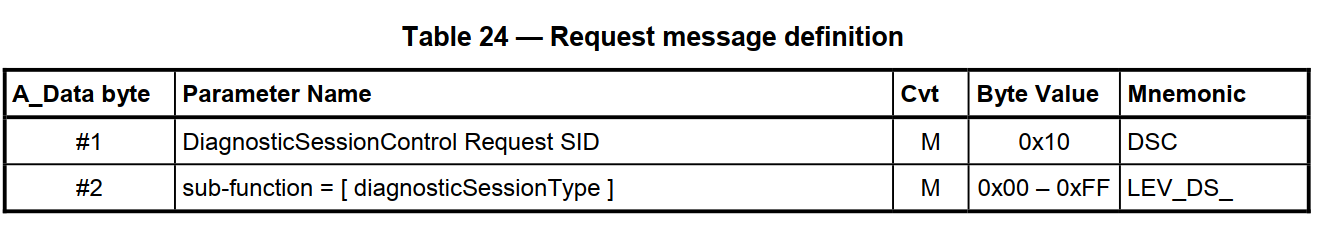






9.2.2 请求消息

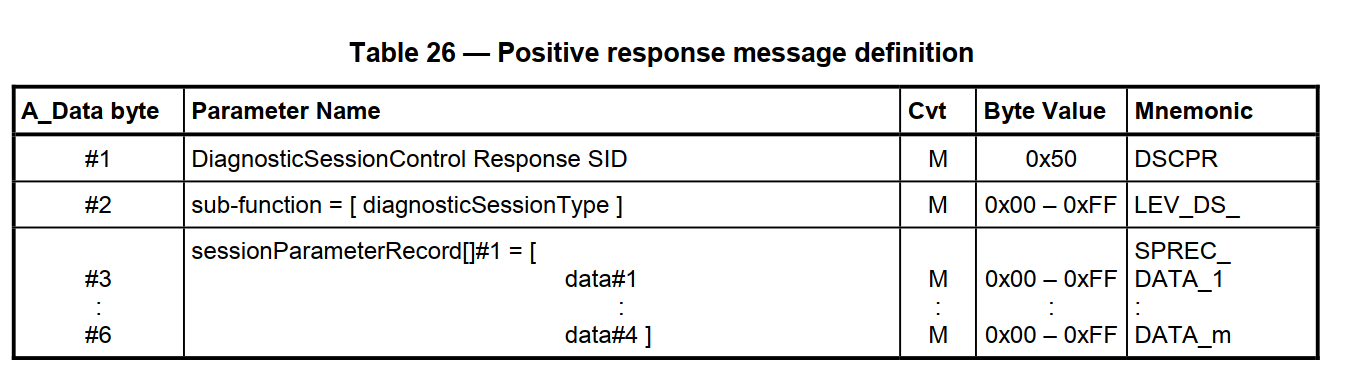
9.2.2.1 请求消息定义



9.2.2.2 请求消息子功能参数级别定义

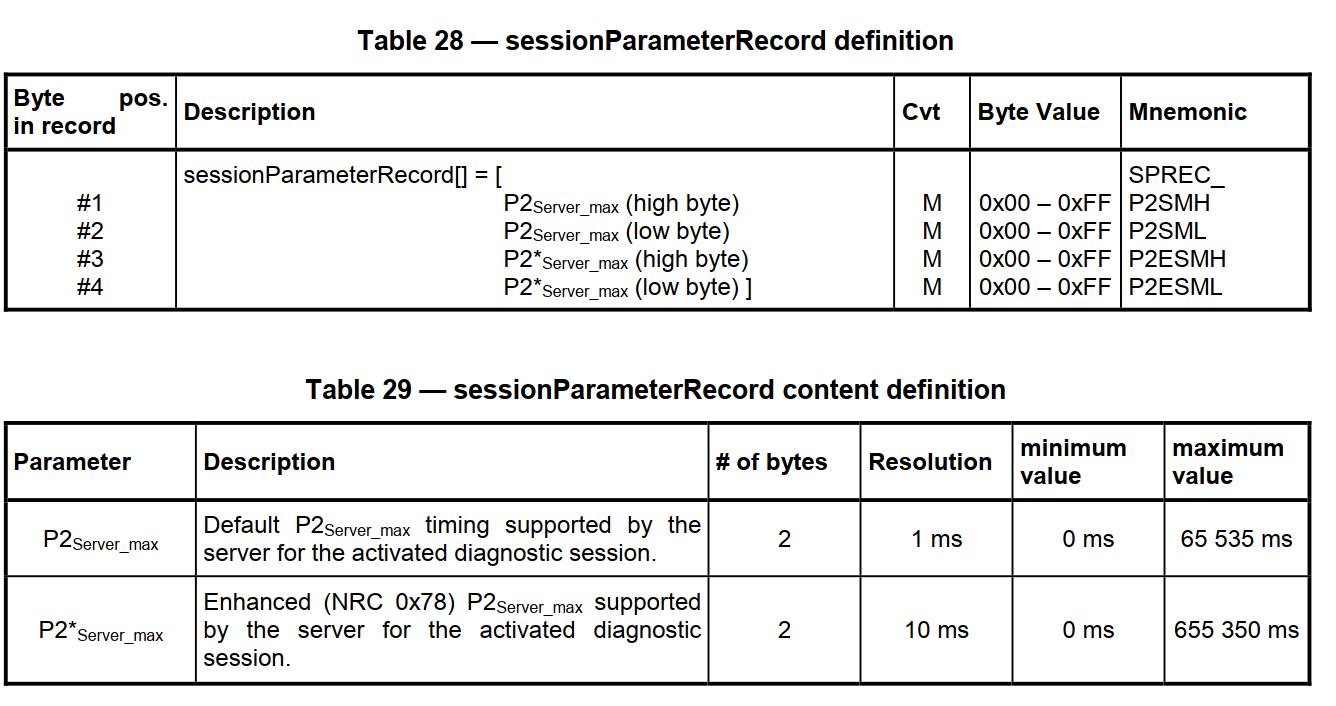
|  |  |  |
| --- | --- | --- |
| Bit6-0 | 描述 | 约定 |
| 0x00 | ISO SAE保留 | 可选 |
| 0x01 | 默认会话层   1. 不需要空闲链接保持 2. 如果当前为默认会话层，有其他会话层请求，如果回复了积极响应，那么要切换到新的会话层；如果回复了消极响应，保持现有会话层。 3. 默认的初始状态在默认会话层 | 可选 |
| 0x02 | 编程层   1. 用于刷写ECU | 可选 |
| 0x03 | 扩展层   1. 用于一些扩展功能的操作的会话层。例如写入信息，执行器测试等。 | 可选 |
| 0x04 | 安全系统诊断会话  使能与安全系统相关功能的会话层。(例如安全气囊) | 可选 |
| 0x05-0x3F | ISO SAE保留 | 强制 |
| 0x40-0x5F | 厂商指定 | 可选 |
| 0x60-0x7E | 系统供应者指定 | 可选 |
| 0x7F | ISO SAE保留 | 强制 |

9.2.3 积极响应



9.2.3.2 积极响应数据参数定义

|  |
| --- |
| diagnosticSessionType  与请求消息的SID一致 |
| sessionParameterRecord  这个参数记录了会话层指定的参数信息，看下表 |



P2Server\_max和P2\*Server\_max参考14229-2。

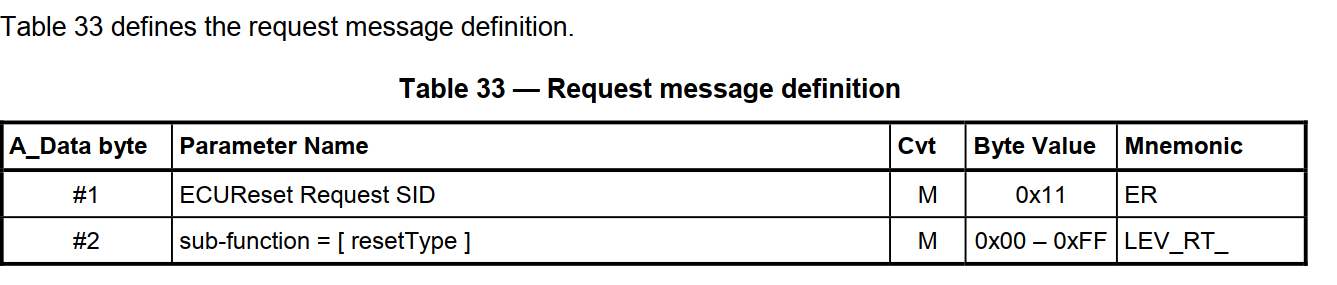
展开释义:

|  |  |
| --- | --- |
| P2Server\_max | 服务端在接收到请求到响应发送之前的最大时间。 |
| P2\*Server\_max | 发送完等待的消极响应(0x78)后到发送下一帧响应的最大时间 |

9.2.4 支持的消极响应代码

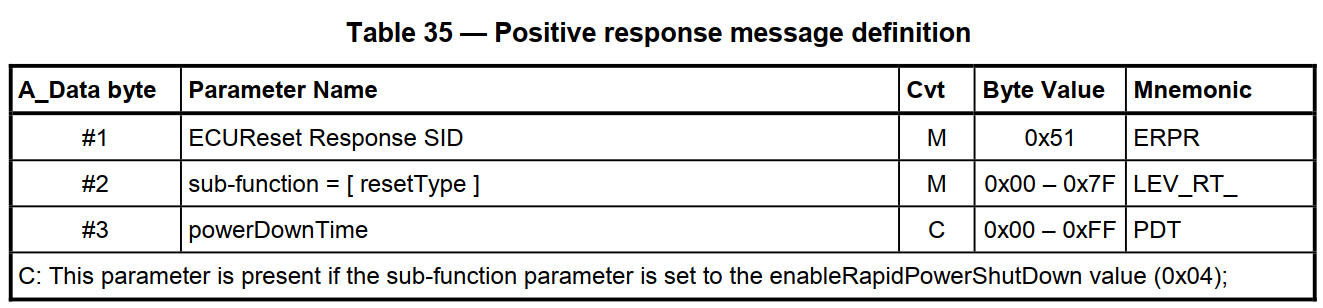
|  |  |
| --- | --- |
| NRC | 描述 |
| 0x12 | 子功能不支持 |
| 0x13 | 错误的消息长度或者无效的格式  例如: 0x10需要2个字节有效长度，而命令中的第一个字节不是02. |
| 0x22 | 条件不正确  例如在此诊断会话层不适合。 |

9.3 ECU复位(0x11)



|  |  |  |
| --- | --- | --- |
| Bit6-0 | 描述 | 约定 |
| 0x00 | ISO SAE保留 | 强制 |
| 0x01 | 硬件复位  它是模拟类型电源断电操作。可能会导致  将易失性存储器和非易失性存储器位置都重新初始化为预定值。 | 可选 |
| 0x02 | 钥匙关复位  此复位类似于司机关闭钥匙操作， 非易失性存储器位置的值被保留，易失性内存将初始化。 | 可选 |
| 0x03 | 软件复位  应用程序复位。不重新初始化先前学习的配置数据等信息。 | 可选 |
| 0x04 | 启用快速关机 该子功能适用于不是由点火驱动而是仅由电池驱动的ECU。因此，关机会强制进入睡眠模式，而不是关闭电源。  该子功能的目的是减少点火开关转到关闭位置后ECU的待机时间。 | 可选 |
| 0x05 | 禁用快速关机 | 可选 |
| 0x06-0x3F | ISO SAE保留 | 强制 |
| 0x40-0x5F | 厂商指定 | 可选 |
| 0x60-0x7E | 系统供应者指定 | 可选 |
| 0x7F | ISO SAE保留 | 强制 |

9.3.3 积极响应



|  |
| --- |
| resetType  已请求服务的子ID相对应 |
| powerDownTime  关机的最小时间。  0x00~0xfe: 0~254s  0xFF: 意味着失败或者时间无效 |

9.3.4 消极响应

|  |  |
| --- | --- |
| NRC | 描述 |
| 0x12 | 子功能不支持 |
| 0x13 | 错误的消息长度或者无效的格式 |
| 0x22 | 条件不正确  当前不适合ECU复位 |
| 0x33 | 安全访问拒绝  未发送安全访问就发送了这条指令 |

9.4 安全访问服务(0x27)

有些服务或者数据不是随意进行访问或者修改，需要通过安全访问，通过安全算法方可进行。

操作流程:

7E0 02 27 01 CC CC CC CC CC(请求种子SEED) 7E8 06 67 01 12 34 56 78 CC(四个随机值)

7E0的ECU根据回复的SEED(12,34,56,78)和预先知道的算法公式(假设算法为颠倒)，生成密钥KEY:

7E0 06 27 02 78 56 34 21 CC(发送密钥KEY) 7E8 02 67 02 CC CC CC CC CC

7E8的ECU根据7E0的KEY和字节回复的种子，进行校验，校验通过，回复积极响应。

请求种子的子功能参数总是一个奇数值，而对应的发送密钥的子功能参数总是这个值加1. 例如（01,02 | 03,04 |05,06）

在任何的时刻都只有一个安全级别起作用。

如果请求某个安全种子时，此安全访问已经通过，此时回复的积极响应，种子的值都设置为0。服务端决不能在未解锁的安全访问的情况下，发送全为0的种子。

车厂可能需要一个延时在错误的尝试发送错误密钥的情况下。

在1个特定的诊断会话层中， 一些诊断功能或者服务可能需要一个成功的安全访问序列。在这种情况下，需要以下的服务序列:

--诊断会话层控制服务

--安全访问服务

--安全的诊断服务

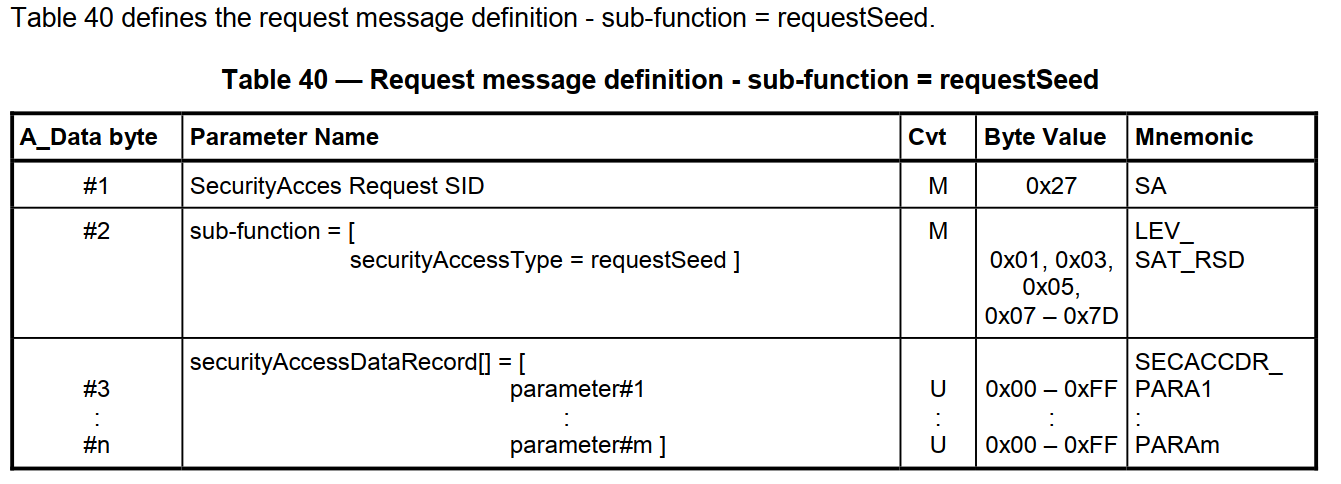
例子:

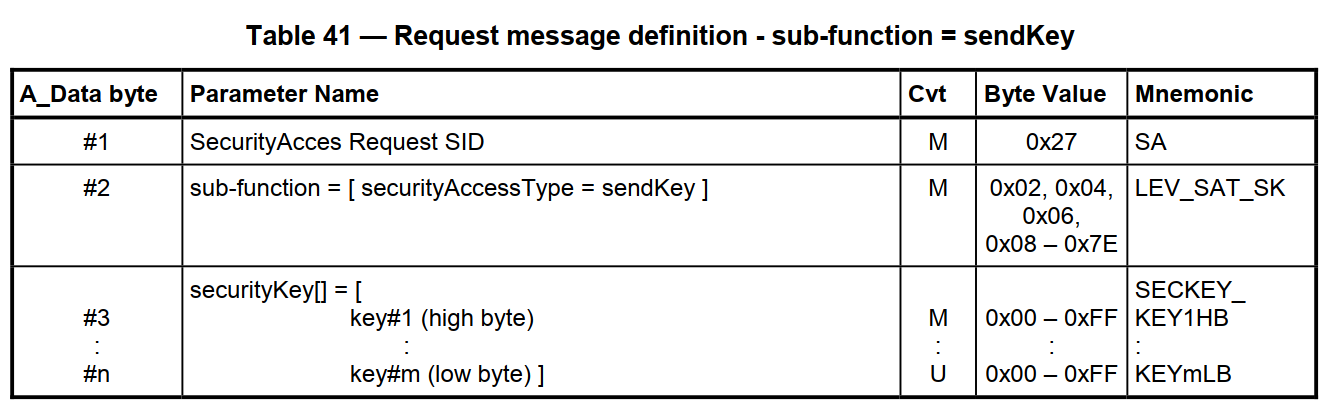
7E0 02 10 03 CC CC CC CC CC 7E8 06 50 03 12 34 45 23 CC

7E0 02 27 01 CC CC CC CC CC 7E8 06 67 01 12 34 56 78 CC

7E0 06 27 02 78 56 34 12 CC 7E8 02 67 02 CC CC CC CC CC

7E0 04 2E 12 32 78 CC CC CC 7E8 01 6E CC CC CC CC CC CC

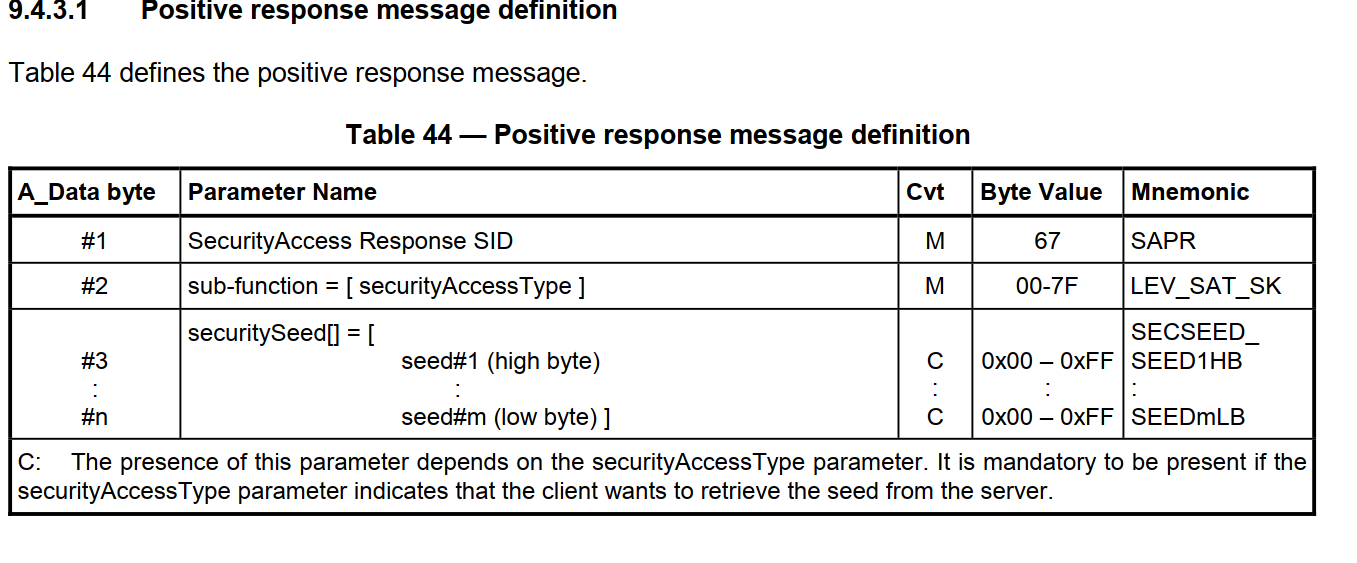




9.4.4.2 子功能参数定义

|  |  |  |
| --- | --- | --- |
| Bit6-0 | 描述 | 约定 |
| 0x00 | ISO SAE保留 | 强制 |
| 0x01 | 请求种子  厂商定义的安全级别 | 可选 |
| 0x02 | 发送密钥  厂商定义的安全级别 | 可选 |
| 0x03,0x05,  0x07-0x41 | 请求种子  厂商定义的安全级别 | 可选 |
| 0x04,0x06,  0x08-0x42 | 发送密钥  厂商定义的安全级别 | 可选 |
| 0x43 – 0x5E | ISO SAE保留 | 强制 |
| 0x5F | ISO26021-2 values  具有为ISO 26021-2中定义的车载烟火设备寿命终止激活定义的不同安全级别的RequestSeed | 可选 |
| 0x60 | ISO26021-2 values | 可选 |
| 0x61 – 0x7E | 系统供应商指定 | 可选 |
| 0x7F | ISO SAE保留 | 强制 |

9.4.3 积极响应



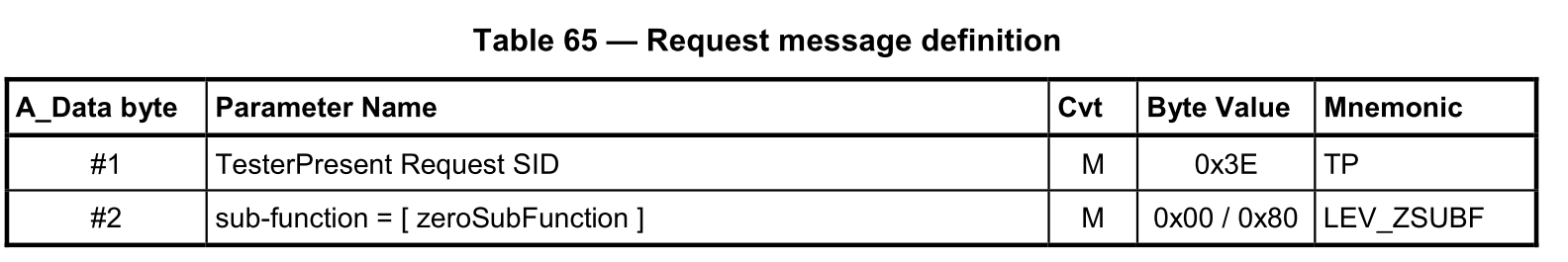
9.4.4 消极响应

|  |  |
| --- | --- |
| NRC | 描述 |
| 0x12 | 子功能不支持 |
| 0x13 | 错误的消息长度或者无效的格式 |
| 0x22 | 条件不正确 |
| 0x24 | 请求序列错误  先发了key，而没有发请求种子指令 |
| 0x31 | 请求超出范围  securityAccessDataRecord保护无效值 |
| 0x35 | 无效KEY  KEY错误 |
| 0x36 | 超过了尝试次数  超过了尝试次数后，会启动延时，此时即便发送正确的种子也是无用的。 |
| 0x37 | 已经启动了延时。  当收到0x36时，再次尝试发送后，会收到0x37. |

9.5 通信控制服务(0x28)

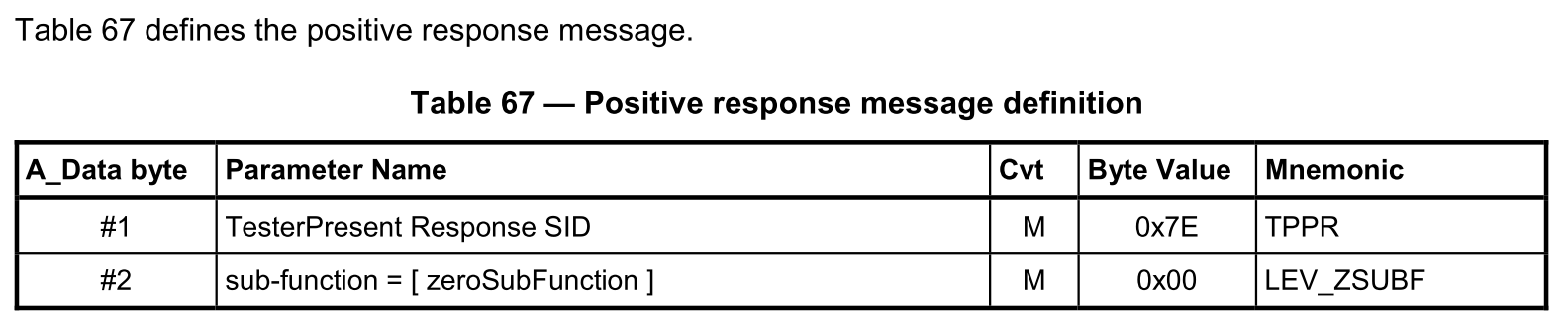
9.6 空闲连接服务(0x3E)

此功能是为了保持在某个会话层中，类似于心跳。（默认会话层不需要空闲连接保持）



|  |
| --- |
| 0x00  此参数值用于指示此服务不支持除抑制正面响应消息指示位之外的任何子功能值 |
| 0x01-0x7E  ISO SAE 保留 |

9.6.3.1 积极响应

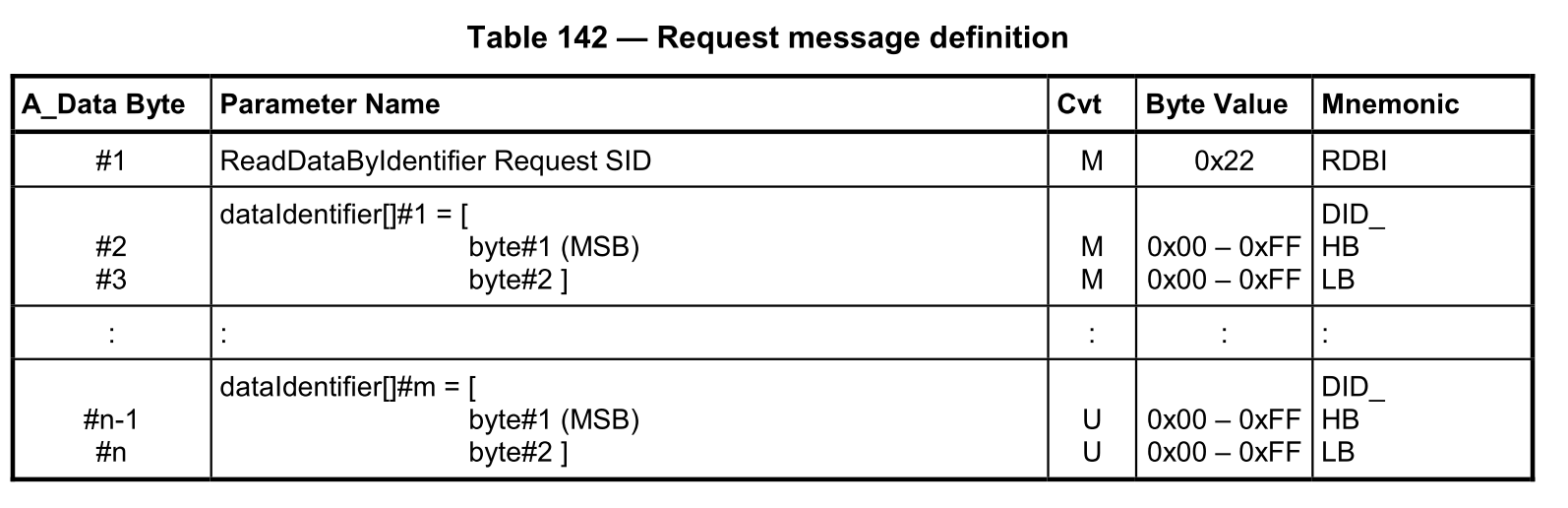


9.6.4 消极响应

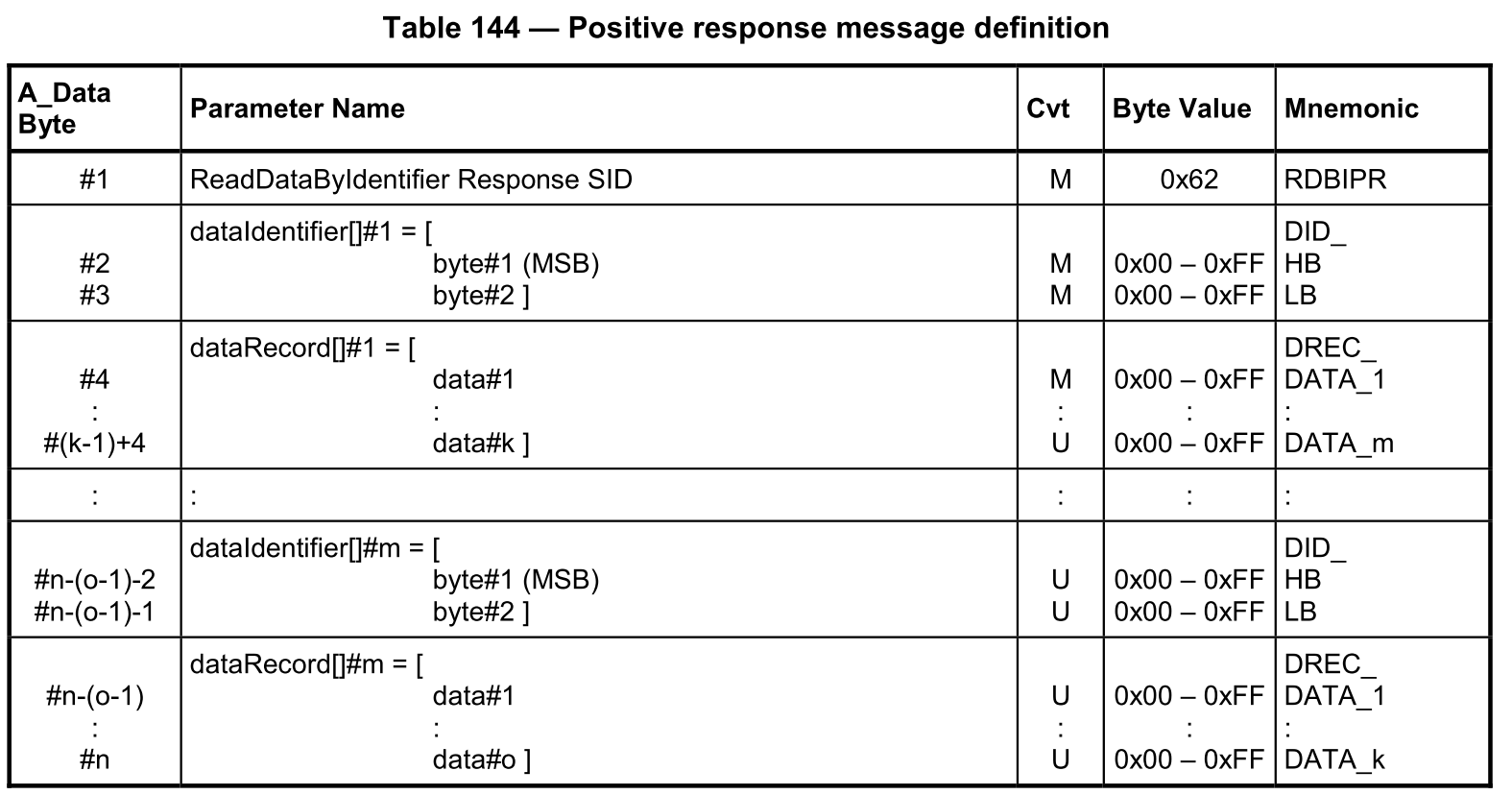
|  |  |
| --- | --- |
| NRC | 描述 |
| 0x12 | 子功能不支持 |
| 0x13 | 错误的消息长度或者无效的格式 |

* 9.7 访问时间参数服务(0x83)
* 9.8 SecuredDataTransmission (0x84) service
* 9.9 ControlDTCSetting (0x85) service
* 9.10 ResponseOnEvent (0x86) service
* 9.11 LinkControl (0x87) service
* 10.2 ReadDataByIdentifier (0x22) service

通过数据标识符读取数据



10.2.3 积极响应



10.2.4 消极响应

|  |  |
| --- | --- |
| NRC | 描述 |
| 0x12 | 子功能不支持 |
| 0x13 | 错误的消息长度或者无效的格式  请求的消息无效或者超出了标识符的最大个数。 |
| 0x14 | 响应太长  超出了传输协议的最大限制 |
| 0x22 | 条件不正确 |
| 0x31 | 请求超出范围  不支持的DID。  当前会话中不支持所请求的dataidentifier;  所请求的动态定义数据标识符尚未被分配; |
| 0x33 | 安全访问拒绝  这个请求的DID需要安全访问服务 |

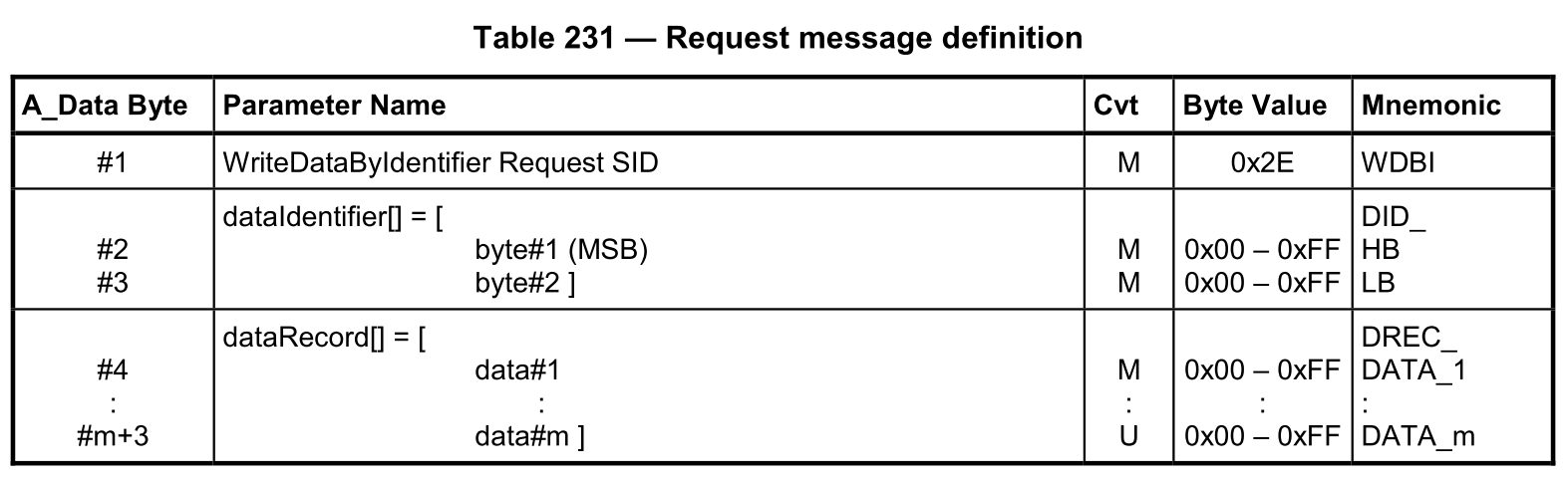
10.3 ReadMemoryByAddress (0x23) service

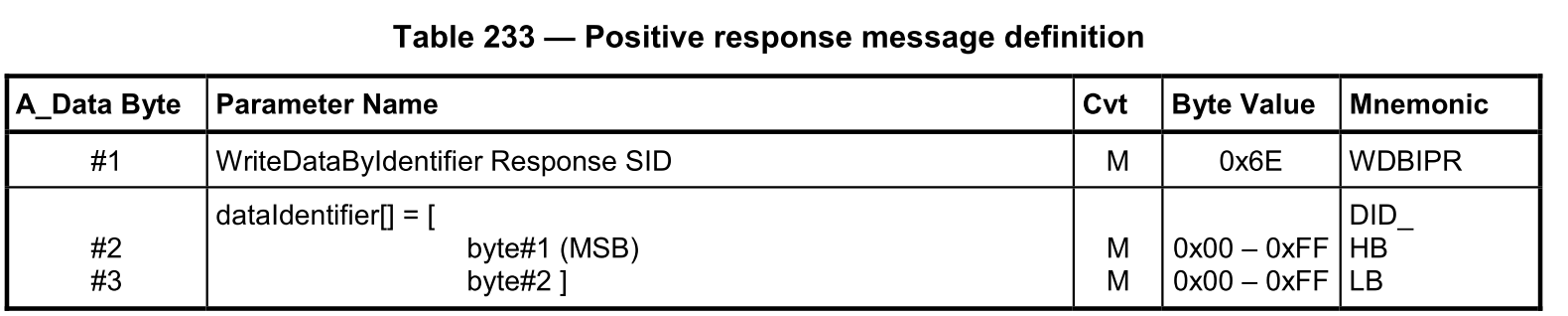
10.4 ReadScalingDataByIdentifier (0x24) service

10.5 ReadDataByPeriodicIdentifier (0x2A) service

10.6 DynamicallyDefineDataIdentifier (0x2C) service

10.7 通过标识符写数据(0x2E)





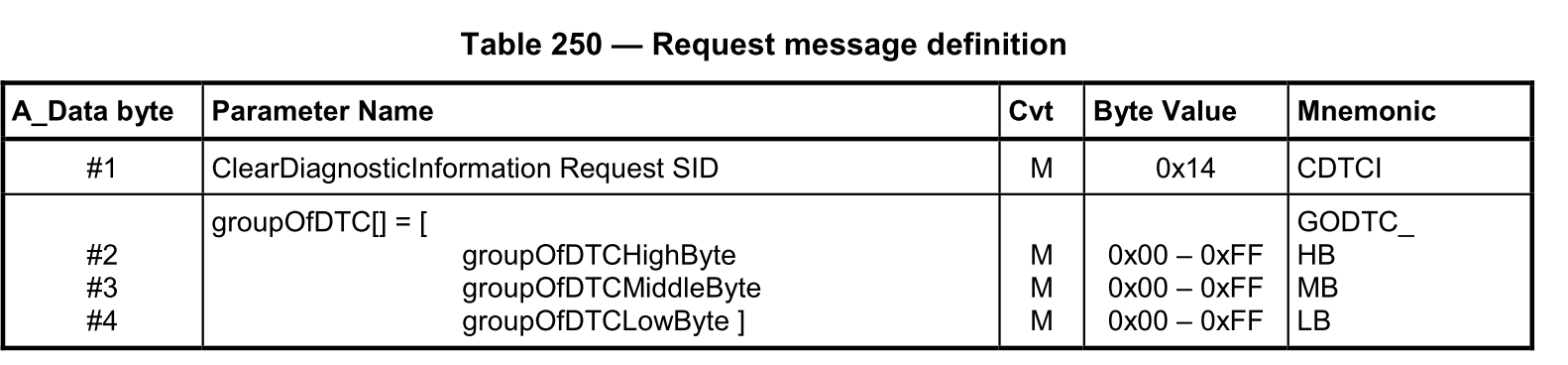
10.7.4 消极响应

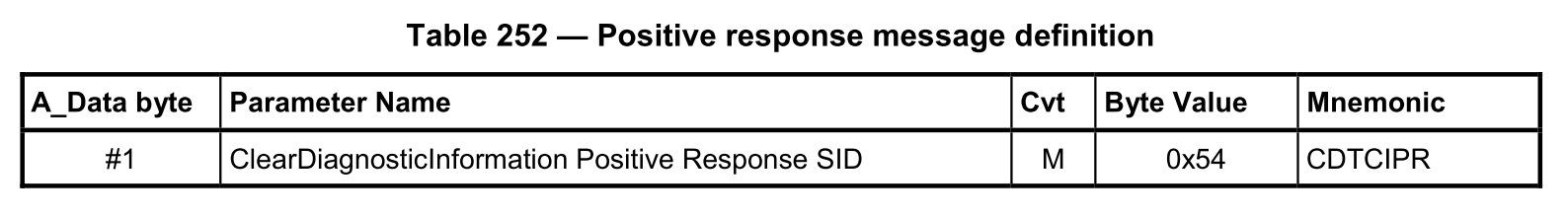
|  |  |
| --- | --- |
| NRC | 描述 |
| 0x13 | 错误的消息长度或者无效的格式 |
| 0x14 | 响应太长  超出了传输协议的最大限制 |
| 0x22 | 条件不正确 |
| 0x31 | 请求超出范围  不支持的DID或者这个DID仅支持读取。  dataIdentifier之后在请求消息中传输的任何数据均无效（如果适用于该节点）； |
| 0x33 | 安全访问拒绝  这个DID需要安全访问服务 |
| 0x72 | 编程失败  写寄存器错误 |

10.8 WriteMemoryByAddress (0x3D) service

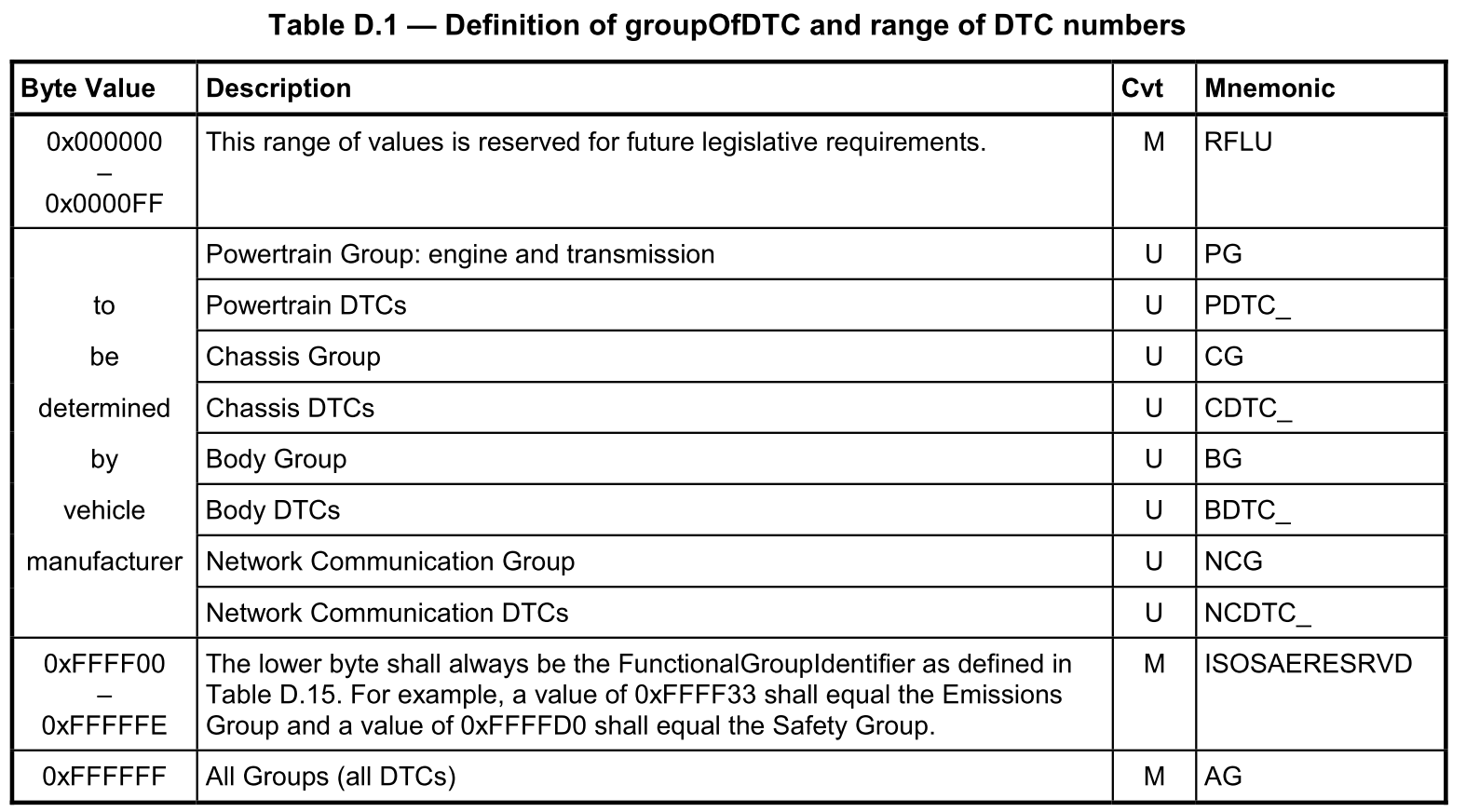
11.2 清除故障码服务(0x14)

在成功清除完故障码后，应该发送一个积极响应。如果当前没有故障码，也应该发送一个积极响应。





|  |
| --- |
| groupOfDTC：  故障码组（例如: 动力，车身，底盘）或者特殊故障码。 |



11.2.4 消极响应

|  |  |
| --- | --- |
| 0x13 | 错误的消息长度或者无效的格式 |
| 0x22 | 条件不正确  内部不允许清除故障信息 |
| 0x31 | 请求超出范围  指定的故障码组不支持 |
| 0x33 | 安全访问拒绝 |
| 0x72 | 编程失败  写寄存器错误 |

11.3 ReadDTCInformation (0x19) Service

1. DTCSnapshot data（DTC闪存数据）： 用于发送故障时，记录的当时与故障相关的数据。

2. DTCExtendedData（DTC扩展的数据）: 从DTC内存或DTC镜像内存中检索与客户定义的DTC和状态掩码组合相关联的DTCExtendedData。DTCExtendedData由与DTC关联的扩展状态信息组成。DTCExtendedData包含在请求时识别的DTC参数值。DTCExtendedData的典型用途是存储与DTC关联的动态数据：

例如: DTC B1故障指示计数器，它表示OBD系统在故障处于活动状态时运行的时间(发动机工作小时数)

DTC发生计数器，计数“测试失败”已报告的驱动循环数，

DTC老化计数器，计数自故障最近失败以来的驾驶循环数，排除测试中未报告“测试通过”或“测试失败”的驾驶循环数 ，

OBD的特定计数器(例如，如果可以在无故障模式下进行循环驾驶，则在“检查引擎”灯关闭前，余下的循环驾驶次数)。

测试通过标准: 服务器/车辆制造商/系统供应商的特定条件，定义被诊断的系统是否在正常、可接受的操作范围内正常运行(例如，无故障存在，被诊断的系统被列为“OK”)。

测试失败标准：服务器/车辆制造商/系统供应商特定的失败条件，这些条件定义了被诊断系统是否未通过测试。

已确认的故障标准：服务器/车辆制造商/系统供应商特定的故障条件，这些条件定义了被诊断的系统是否确实存在问题（已确认），并保证将DTC记录存储在长期内存中。

发生计数器：由某些服务器维护的计数器，用于记录给定DTC测试报告唯一的测试失败发生的实例数。

老化：某些服务器评估每个内部诊断的过去结果以确定是否可以从长期存储中清除已确认DTC的过程，例如如果校准了无故障循环次数。

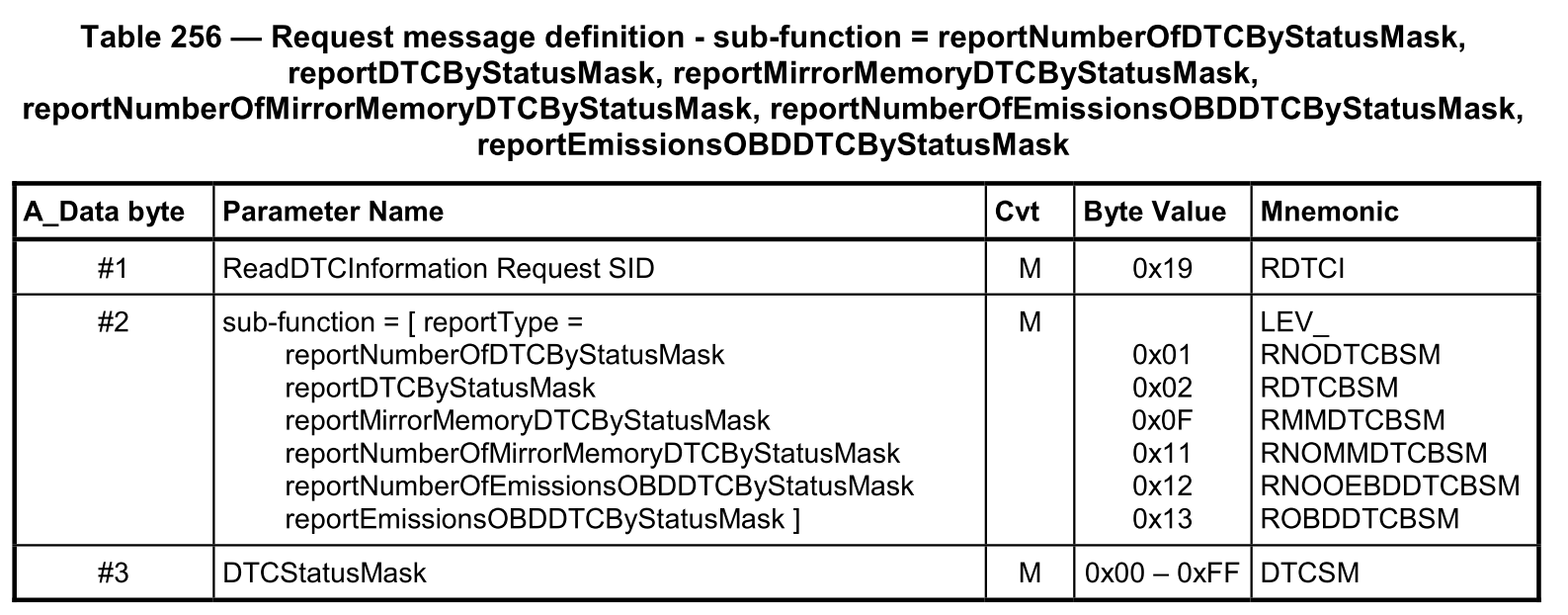
在读取DTC信息的肯定响应中，给定DTC值（例如0x080511）不得重复报告一次，但读取DTCSnapshotRecords除外，该响应可能包含同一DTC的多个DTCSnapshotRecords。

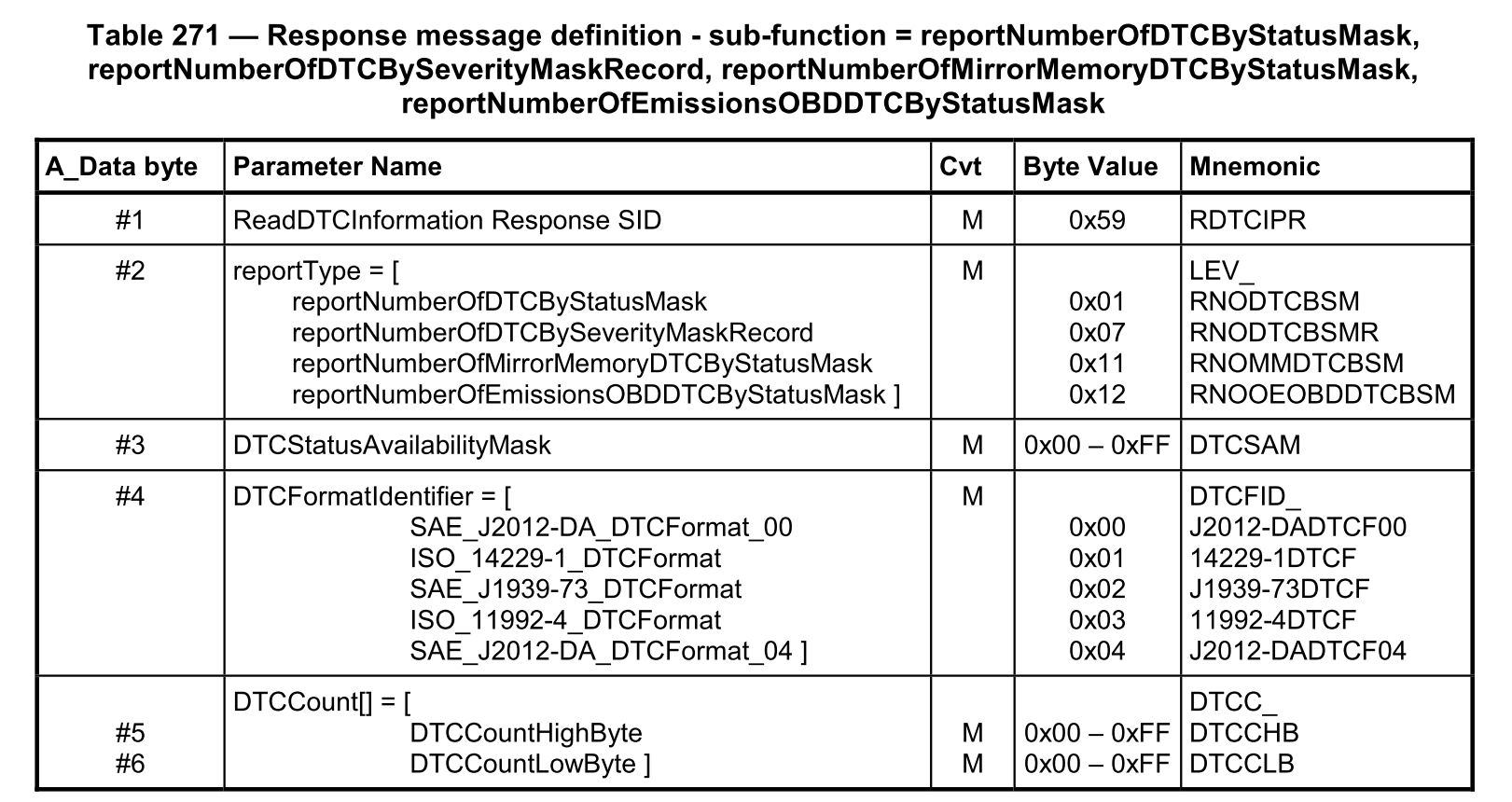
使用分页缓冲区处理读取DTC时（尤其是对于子功能= reportDTCByStatusMask），创建响应时DTC的数量可能会减少。在这种情况下，响应中应填充DTC 0x000000和DTC状态0x00。客户应将这些DTC视为在响应消息中不存在。

11.3.1.2 检索与客户端定义的状态掩码匹配的DTC数量（子功能= 0x01 reportNumberOfDTCByStatusMask）---读取故障码个数

客户端可以通过发送对此服务的请求并将子功能设置为reportNumberOfDTCByStatusMask，来检索与客户端定义的状态掩码匹配的DTC数量。对这个请求的响应包含DTCStatusAvailabilityMask，它提供服务器支持用于屏蔽目的的DTC状态位的指示。在DTCStatusAvailabilityMask之后，响应包含DTCFormatIdentifier，该报告报告有关DTC格式和编码的信息。 DTCFormatIdentifier之后是DTCCount参数，该参数是2字节无符号数字，其中包含基于客户端提供的状态掩码的服务器内存中可用的DTC数量。

子功能reportNumberOfMirrorMemoryDTCByStatusMask具有与子功能reportNumberOfDTCByStatusMask相同的功能，不同之处在于它从DTC镜像存储器中返回DTC的数量。



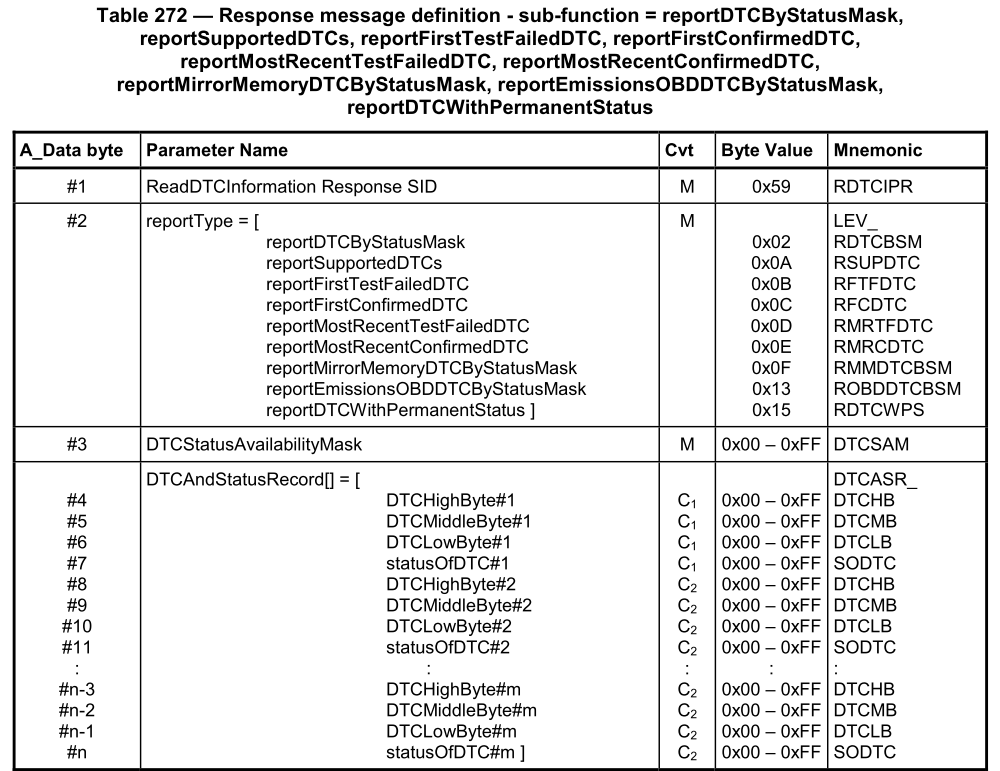


11.3.1.3 检索与客户端定义的状态掩码匹配的DTC列表（子功能= 0x02 reportDTCByStatusMask）---读取故障码

客户端可以通过发送带有设置为reportDTCByStatusMask的子功能字节的请求来检索满足客户端定义的状态掩码的DTC列表。此子功能允许客户端请求服务器报告“ testFailed”或“ confirmed”或“ etc”的所有DTC。

评估应按以下方式进行：服务器应在客户端请求中指定的掩码与服务器支持的每个DTC关联的实际状态之间执行按位逻辑与运算。除了DTCStatusAvailabilityMask，服务器还应返回所有与运算结果非零（即（statusOfDTC＆DTCStatusMask）！= 0）的DTC。如果客户端指定的状态掩码包含服务器不支持的位，则服务器应仅使用其支持的位来处理DTC信息。如果服务器中没有DTC符合客户端请求中指定的屏蔽标准，则在肯定响应消息中的DTCStatusAvailabilityMask字节之后，不得提供DTC或状态信息。

在客户端成功发出ClearDiagnosticInformation请求后，应清除DTC状态信息（有关在服务器中接收到ClearDiagnosticInformation服务请求的情况下DTC状态位处理的更多说明，请参见D.2中的DTC状态位定义）。

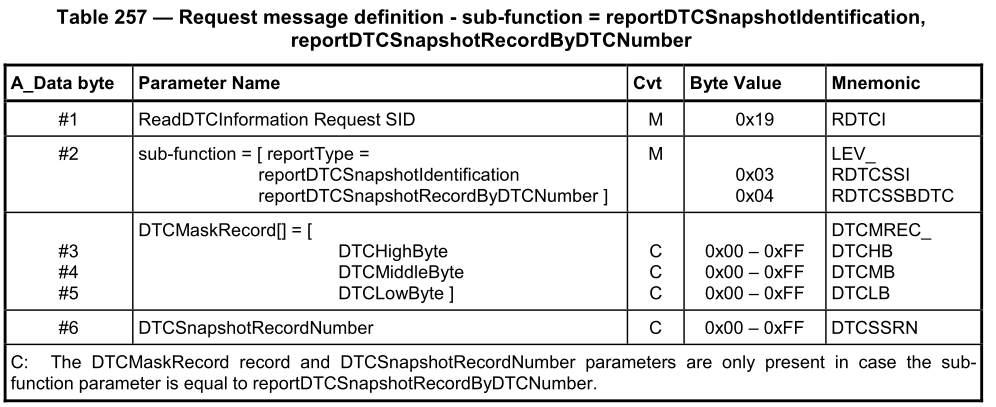


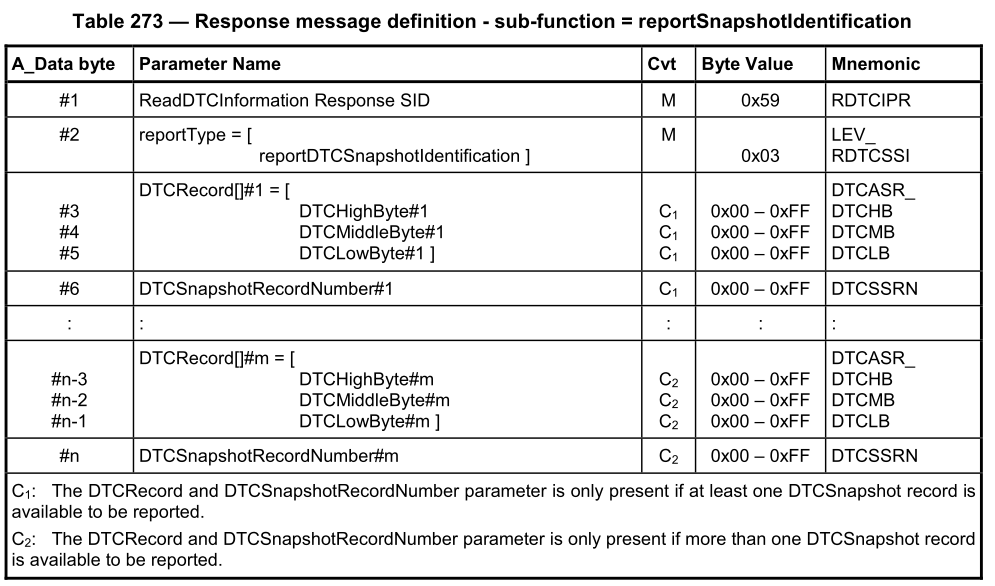
11.3.1.4 检索DTCSnapshot记录标识（子功能= 0x03 reportDTCSnapshotIdentification）

客户端可以通过发送对此服务的请求并将子功能设置为reportDTCSnapshotIdentification来检索所有捕获的DTCSnapshot记录的DTCSnapshot记录标识信息。服务器应返回所有已存储DTCSnapshot记录的DTCSnapshot记录标识信息列表。服务器在响应消息中为单个DTCSnapshot记录放置的每个项目都应包含一个DTCRecord（包含DTC编号（高，中，低字节））和DTCSnapshot记录编号。如果为单个DTC存储了多个DTCSnapshot记录，则服务器应为每个事件在响应中放置一项，并为每个事件使用不同的DTCSnapshot记录号（用于以后的记录数据检索）。

注意服务器可能支持为单个DTC存储多个DTCSnapshot记录，以跟踪每次DTC出现时出现的状况。该功能的支持，“发生”标准的定义以及要支持的DTCSnapshot记录的数量应由系统供应商/车辆制造商定义。

客户端成功发出ClearDiagnosticInformation请求后，应清除DTCSnapshot记录标识信息。车辆制造商有责任为内存溢出（存储在服务器中的已存储DTC和DTCSnapshot数据的内存空间）指定删除已存储DTC和DTCSnapshot数据的规则。





11.3.1.5 检索客户端定义的DTC掩码的DTCSnapshot记录数据（子功能= 0x04 reportDTCSnapshotRecordByDTCNumber）

客户端可以通过发送对该服务的请求并将子功能设置为reportDTCSnapshotRecordByDTCNumber，来检索客户端定义的DTCMaskRecord的捕获的DTCSnapshot记录数据以及DTCSnapshot记录号。服务器应在其支持的DTC中搜索与客户端指定的DTCMaskRecord完全匹配（包含DTC编号（高，中，低字节））。客户请求中提供的DTCSnapshotRecordNumber参数应指定DTCSnapshot记录数据正在请求中的特定DTC的特定出现。

注1：DTCSnapshotRecordNumber与DTCStoredDataRecordNumber共享的地址空间不同。

如果服务器支持为单个DTC存储多个DTCSnapshot记录的功能（此功能的支持由系统供应商/车辆制造商定义），则建议服务器还实现reportDTCSnapshotIdentification子功能参数。建议客户端首先通过reportDTCSnapshotRecordByDTCNumber请求请求特定的DTCSnapshotRecordNumber，然后再请求使用子功能参数reportDTCSnapshotIdentification存储的DTCSnapshot记录的标识。

还建议支持子功能参数reportDTCSnapshotRecordIdentification，以使客户端有机会直接标识存储的DTCSnapshot记录，而不是通过解析服务器的所有存储的DTC来确定是否存储了DTCSnapshot记录。

系统供应商/车辆制造商应负责定义在此类服务器中捕获的DTCSnapshot记录是否存储与故障发生信息相关的数据作为ECU文档的一部分。

如果已识别出客户端定义的DTCMaskRecord和DTCSnapshotRecordNumber参数（DTCSnapshotRecordNumber参数不等于0xFF）失败，则服务器应连同DTC编号和statusOfDTC一起返回单个预定义的DTCSnapshotRecord。

注2：确切的失效标准应由系统供应商/车辆制造商规定。

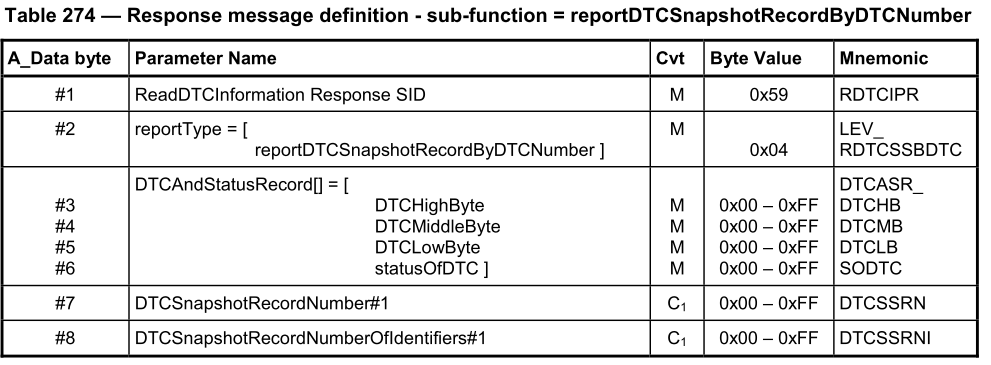
DTCSnapshot记录可能包含多个数据参数，这些数据参数可用于在故障发生时重建车辆状况（例如B +，RPM，时间戳）。

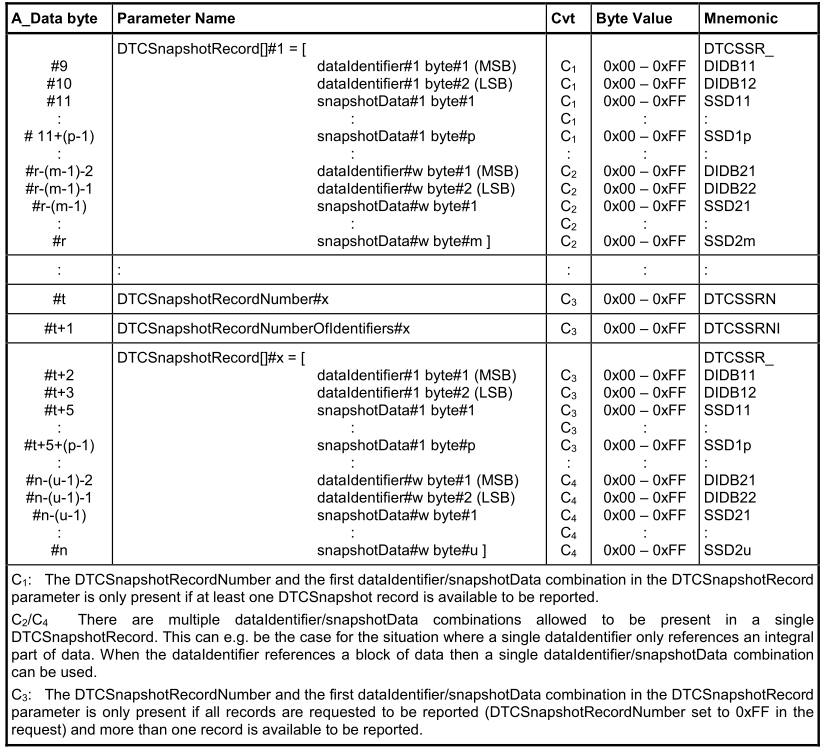
车辆制造商应定义DTCSnapshotRecord的格式和内容。首先在DTCSnapshotRecord中报告的数据包含一个dataIdentifier，用于标识随后的数据。可以在DTCSnapshotRecord中重复此dataIdentifier /数据组合。在DTCSnapshotRecord中使用一个或多个dataIdentifiers可以为单个DTC存储不同类型的DTCSnapshotRecords，以用于发生不同的故障。每个DTCSnapshotRecord都应提供一个参数，该参数指示每个DTCSnapshotRecord中包含的记录DataIdentifier的数目，以辅助数据检索。

服务器应在单个响应消息中报告一个DTCSnapshot记录，除非客户端已将DTCSnapshotRecordNumber设置为0xFF，因为这将导致服务器在单个响应消息中响应为客户端定义的DTCMaskRecord存储的所有DTCSnapshot记录。 DTCAndStatusRecord仅在响应消息中包含一次。如果客户端在其请求中已使用0xFF作为参数DTCSnapshotRecordNumber，则服务器应以数字升序报告为特定DTC捕获的所有DTCSnapshot记录。

如果客户端指定的DTCMaskRecord或DTCSnapshotRecordNumber参数无效或服务器不支持，则服务器应作出否定响应。这与客户端指定的DTCMaskRecord和/或DTCSnapshotRecordNumber参数确实有效并且受服务器支持但没有与之关联的DTCSnapshot数据的情况有所不同（例如，因为从未发生过指定事件的失败事件） DTC或记录号）。服务器应发送仅包含DTCAndStatusRecord（所请求的DTC编号（高，中，低字节）的回声加上statusOfDTC）的肯定响应。

客户端成功发出ClearDiagnosticInformation请求后，应清除DTCSnapshot信息。车辆制造商有责任在内存溢出（存储在服务器中的存储的DTC和DTCsnapshot数据的内存空间）的情况下，指定删除存储的DTC和DTCSnapshot数据的规则。





11.3.1.6 Retrieving DTCStoredData record data for a client defined record number (sub-function = 0x05 reportDTCStoredDataByRecordNumber)

12.2 InputOutputControlByIdentifier (0x2F) service

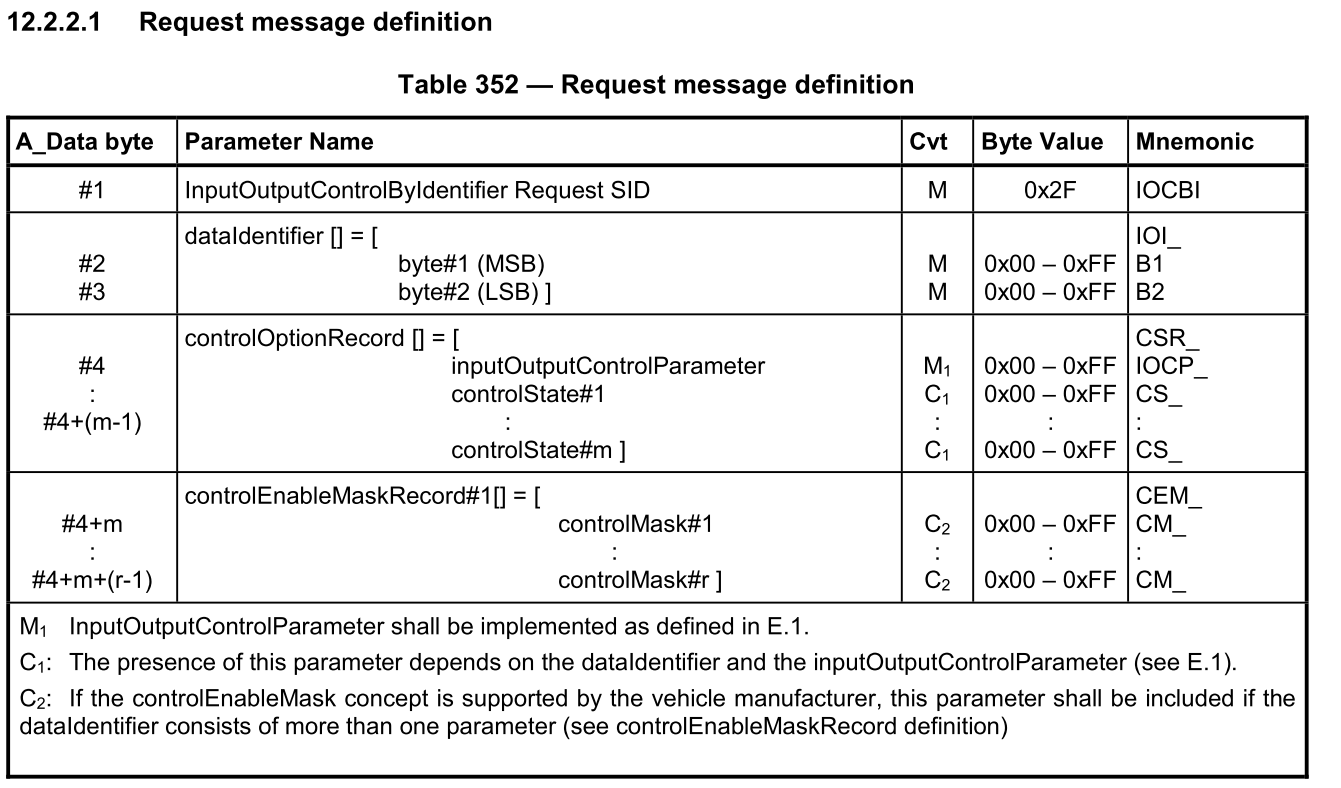
客户端使用InputOutputControlByIdentifier服务将输入信号，内部服务器功能和强制控制的值替换为电子系统的输出（执行器）的值。通常，此服务用于相对简单（例如，静态）的输入替换/输出控制，而如果需要更复杂的输入替换/输出控制，则使用例程控制。

客户端请求消息包含一个dataIdentifier，用于引用服务器的输入信号，内部服务器功能和/或输出信号（执行器）（在设备控制访问的情况下，它可能引用一组信号） 。 controlOptionRecord参数应包含服务器的输入信号，内部功能和/或输出信号所需的所有信息。如果要控制的数据标识符引用多个参数（即，数据标识符被打包或位图映射），则车辆制造商可能要求请求消息中包含controlEnableMask。如果车辆制造商选择支持EnableMask概念，则此服务的所有类型InputOutputControlByIdentifier请求都必须具有controlEnableMask参数。如果在引用已测量的输出值或反馈值的dataIdentifier上请求inputOutputControlByIdentifier，则服务器应负责在服务器控制策略内替换正确的目标值，以便普通服务器控制策略将尝试从客户端达到所需状态请求消息。

如果请求控件已成功启动或已达到其所需状态，则服务器应发送肯定响应消息。即使dataIdentifier当前不在测试人员的控制下，服务器也应使用returnControlToECU的inputOutputControlParameter向请求消息发送肯定响应消息。另外，服务器在接收到returnControlToECU请求时，应始终为客户端提供将controlMask（如果支持）位全部设置为“ 1”的功能，以便将对打包或位映射的数据标识符的控制权完全返回给ECU。在请求消息的controlOptionRecord参数内，inputOutputControlParameter之后的controlState字节的格式和长度应与所请求的dataIdentifier的dataRecord的长度和格式完全匹配。这样，应确保可以使用具有相同DID的服务ReadDatabyIdentifier检索实际的输出或输入状态。

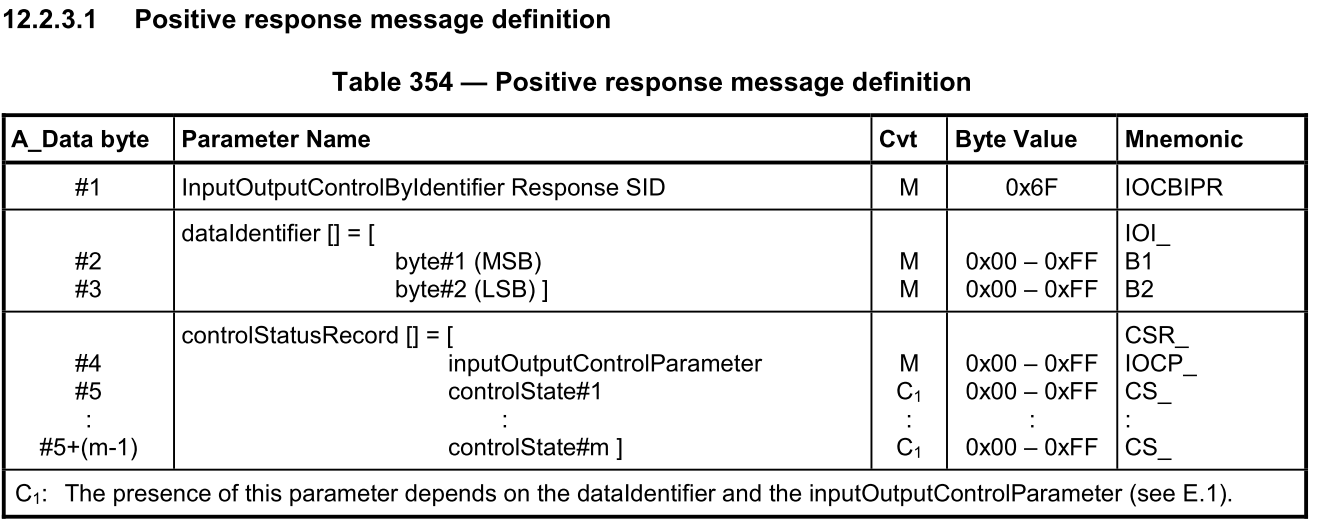
当使用inputOutputControlByIdentifier服务执行输入替换或输出控制时，ECU接受请求有两个基本要求。第一个是从所有上游控制策略中断开dataIdentifier中参数引用的适当数据对象，否则所有上游控制策略都会更新数据对象值。第二种是将值替换为适当的数据对象，这些数据对象将用于控制策略的所有下游活动。例如，测试人员要求直接强行接通前照灯将需要防止前照灯开关位置影响前照灯输出，并将所需的“开”状​​态代入最终决定前照灯的功能所使用的数据对象中说明所需的输出。

该服务允许在单个请求消息中控制单个dataIdentifier及其对应的参数。这样做，服务器将以单个响应消息进行响应，其中包括请求消息的dataIdentifier以及controlStatus信息。



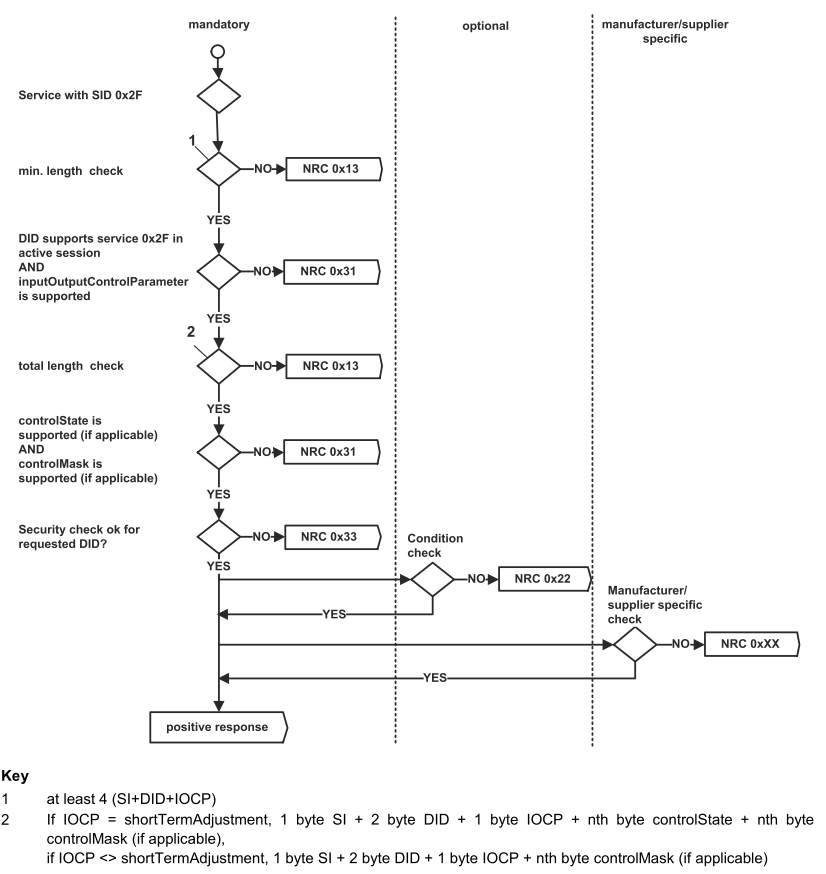
12.2.2.2 子功能参数定义

|  |
| --- |
| dataIdentifier : 信号的标识符，定义的范围在C.1中 |
| controlOptionRecord: 具体内容在E.1 |
| controlEnableMaskRecord:  它包含一个或多个字节（controlMask＃1至controlMask＃r）。仅当要控制的dataIdentifier包含多个参数（即dataIdentifier定义为位映射或打包）时，才应支持controlEnableMaskRecord。 controlEnableMaskRecord中应有一位与dataIdentifier中定义的每个单独参数相对应。当要控制的dataIdentifier仅包含单个参数时，不支持controlEnableMaskRecord。  注意dataIdentifier中的每个参数可以是任意数量的位。  controlEnableMaskRecord中每个位的值应确定dataIdentifier中的相应参数是否会受到请求的影响。 controlEnableMaskRecord中的比特值“ 0”应表示相应参数不受此请求的影响，而比特值“ 1”应表示相应参数不受此请求的影响。 ControlMask＃1的最高有效位应对应于ControlState中的第一个参数，从ControlState＃1的最高有效位开始，ControlMask＃1的第二个最高有效位应与ControlState＃2中的第二个参数相对应，并继续以这种方式利用必要的ControlMask字节来屏蔽所有参数。例如，ControlMask＃2的最低有效位将对应于controlState中的第16个参数。对于位图数据标识符，不受支持的位还应在controlEnableMaskRecord中具有相应的位，以便controlEnableMaskRecord中每个参数的掩码位的位置应与controlState中相应参数的位置完全匹配。 |
|  |



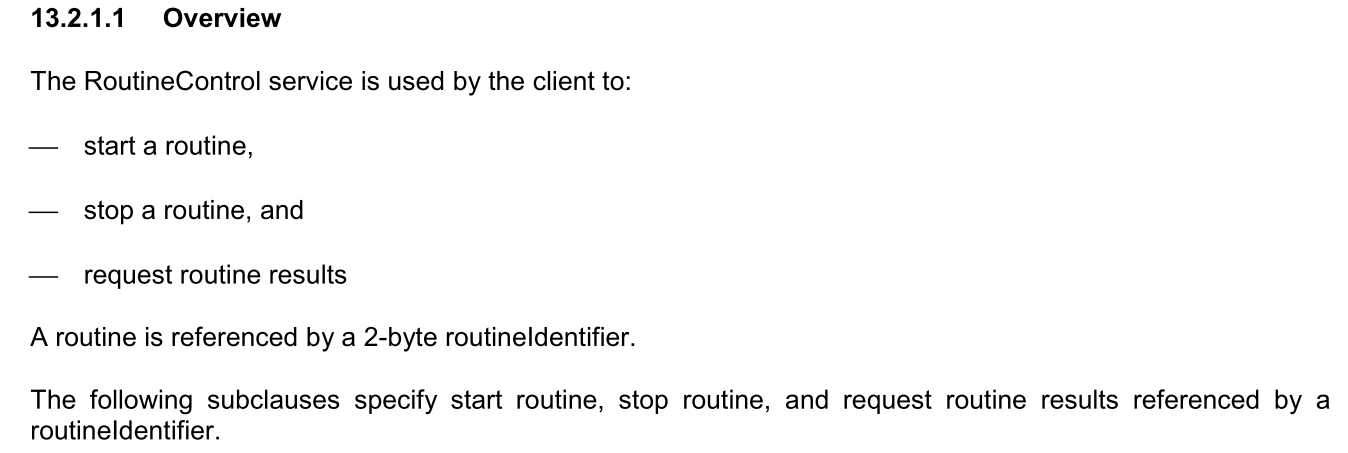
12.2.4 消极响应代码

|  |  |
| --- | --- |
| 0x13 | 错误的消息长度或者无效的格式 |
| 0x22 | 条件不正确 |
| 0x31 | 请求超出范围   1. DID不支持 2. 参数值无效 3. 控制状态无效 4. ControlEnableMaskRecord中位不支持 |
| 0x33 | 安全访问拒绝 |



13.2 RoutineControl (0x31) service

客户端使用RoutineControl服务执行已定义的步骤序列并获取任何相关结果。该服务具有很大的灵活性，但是典型的用法可能包括诸如擦除内存，重置或学习自适应数据，运行自检，覆盖常规服务器控制策略以及控制服务器值随时间变化等功能。预定义的序列（例如，封闭的敞篷车顶）仅举几例。通常，当用于控制输出时，此服务用于更复杂的类型控制，而inputOutputControlByIdentifier用于相对简单（例如，静态）的输出控制。



13.2.1.2 开始

如果响应消息是肯定的或否定的，则表明该请求已被执行或正在执行中，例程应在StartRoutine请求消息完成与第一响应消息完成之间的某个时间在服务器内存中启动。

例程可以是运行而不是正常操作代码的测试，也可以是在正常操作代码运行的情况下启用和执行的例程。特别是在第一种情况下，可能有必要在使用StartRoutine服务之前，使用DiagnosticSessionControl服务在特定的诊断会话中切换服务器，或者使用SecurityAccess服务解锁服务器。

13.2.1.3 停止

在完成StopRoutine请求消息和完成第一条响应消息后，如果响应消息是肯定的还是否定的，则表示服务器例程应在服务器的内存中停止，这表明响应例程已停止执行或正在执行。要执行的进度。

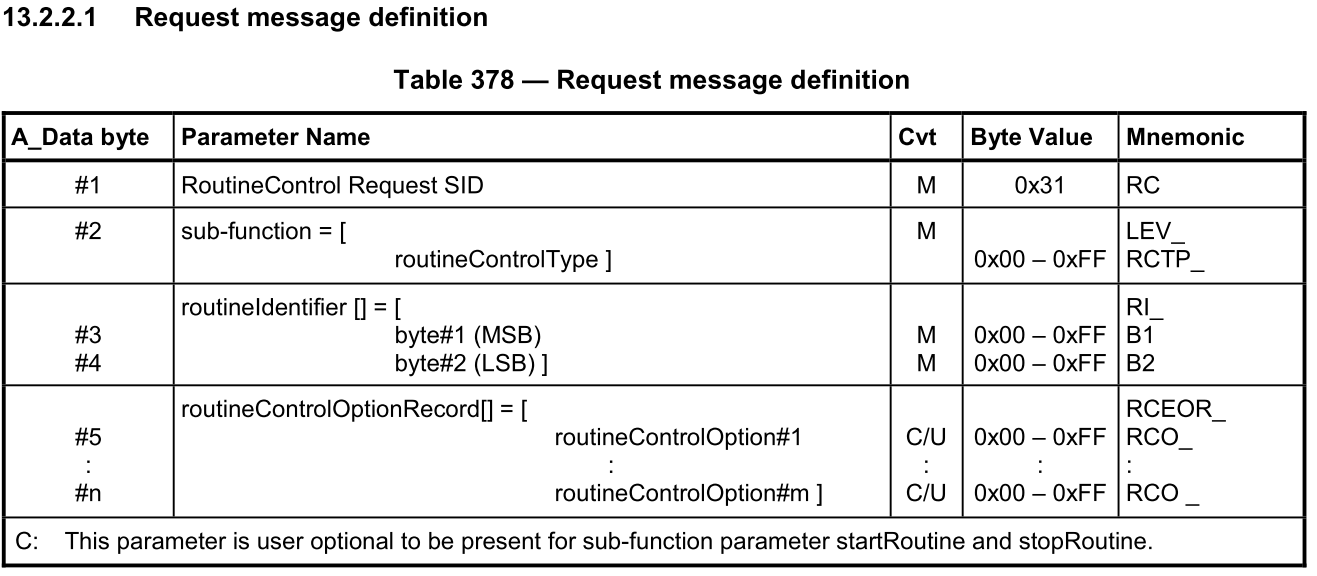
注意-服务器例程应在服务器存储器中按编程或先前初始化的任何时间停止。

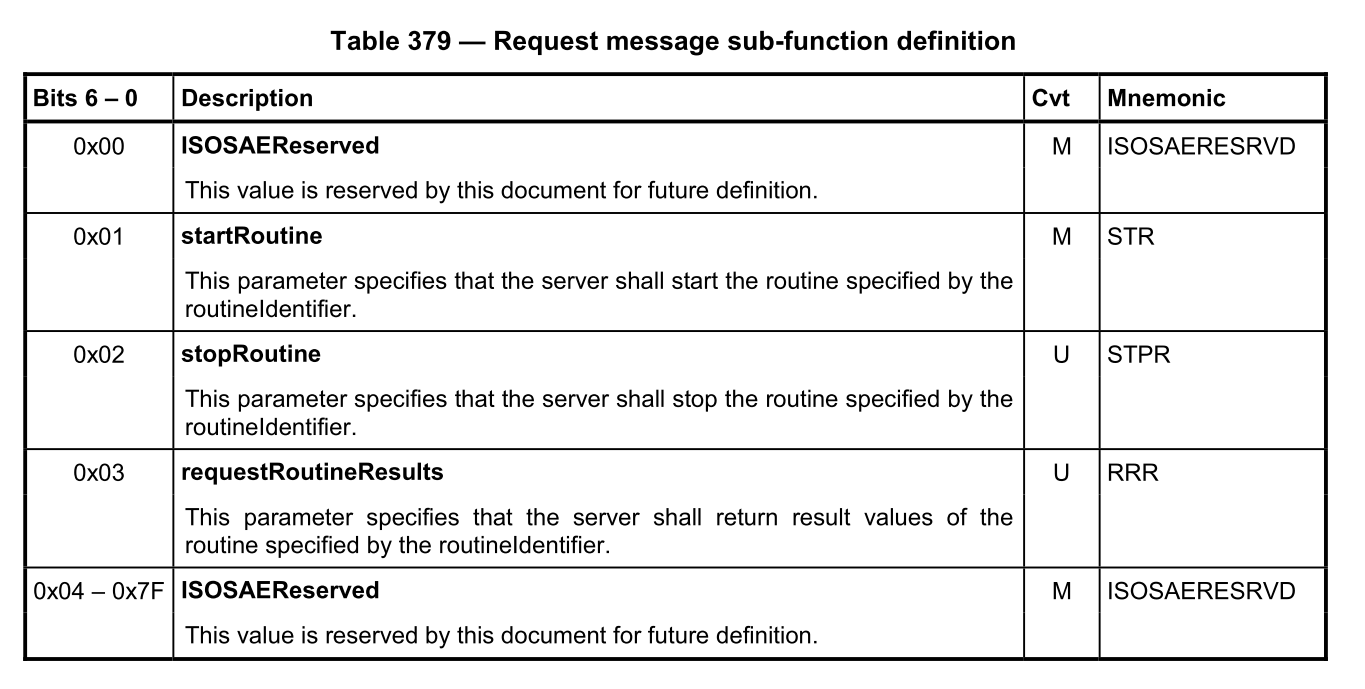
13.2.1.4 请求结果

客户端使用此子功能来请求由例程标识符标识并由在服务器的内存中执行的例程生成的结果（例如退出状态信息）。

根据例行结果（可能已在stopRoutine子功能参数的肯定响应消息中接收到）（例如，正常/异常Exit With Results），应使用requestRoutineResults子功能。

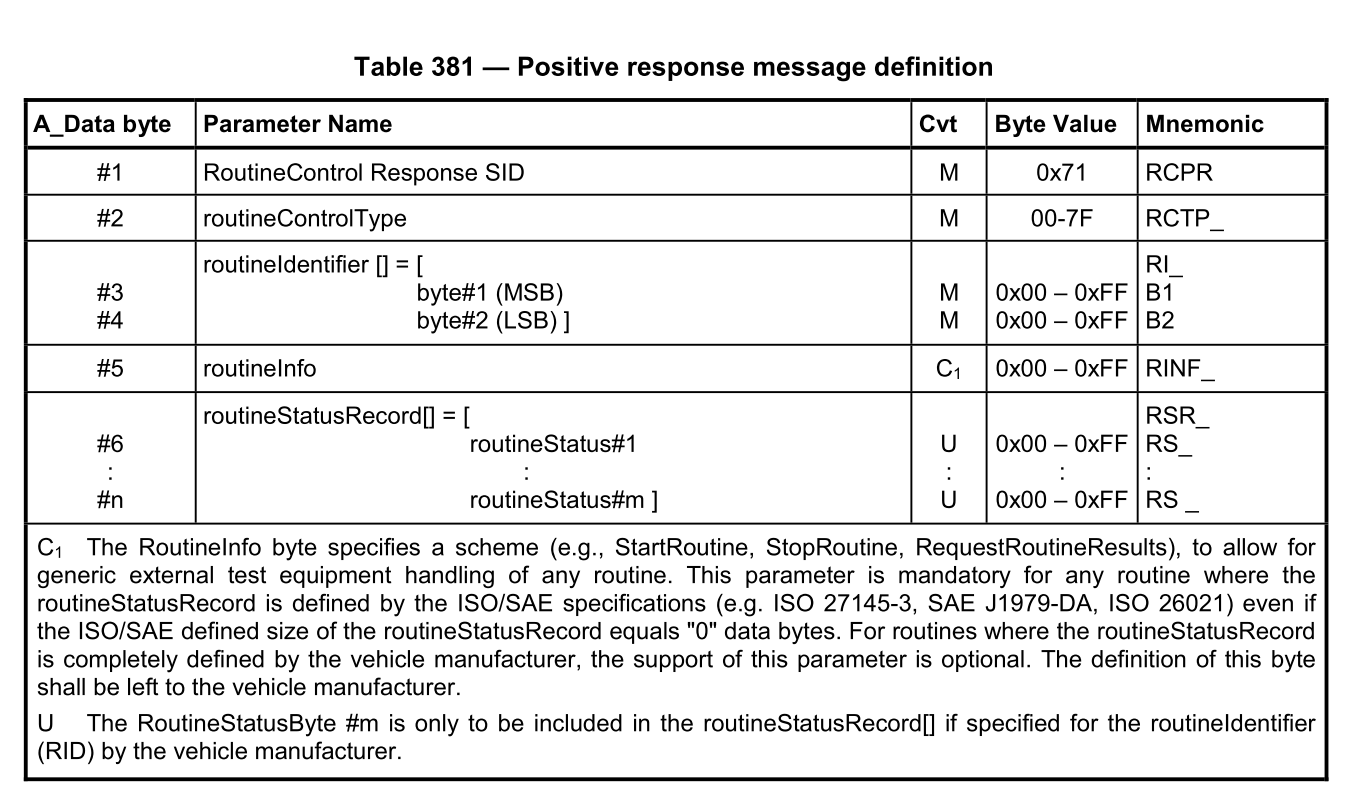
例程结果的一个示例可能是服务器收集的数据，由于服务器性能的限制，在例程执行期间无法传输这些数据。





13.2.2.3 请求消息数据参数定义

|  |
| --- |
| routineIdentifier : 范围看F.1 |
| routineControlOptionRecord: 例程进入选项参数，可以选择指定例程的开始条件（例如timeToRun，startUpVariables等），或者可以选择退出例程的退出选项参数，例如可以指定例程的停止条件（例如timeToExpireBeforeRoutineStops，变量等）。 |

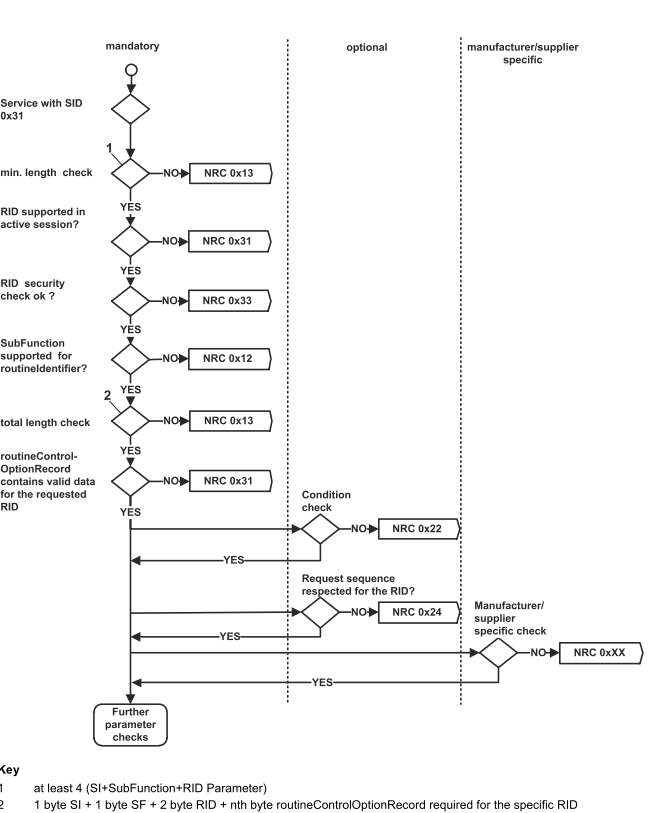


13.2.3.2 积极响应消息数据参数定义

|  |
| --- |
| routineControlType: 与发送消息的routineControlType相同 |
| routineIdentifier: 与发送消息的routineIdentifier相同 |
| routineInfo：RoutineInfo字节编码是特定于车辆制造商的，并为车辆制造商提供了一种机制，以基于此返回值支持所有已实施例程的通用外部测试设备处理（例如，如果需要stopRoutine或requestRoutineResults）。 |
| routineStatusRecord:  此参数记录用于提供给客户端：  例程启动后有关服务器状态的其他信息或例程停止后有关服务器状态的其他信息（例如，总运行时间，例程在停止前生成的结果等）或结果（例程的退出状态信息），该例程先前已在服务器中停止。 |

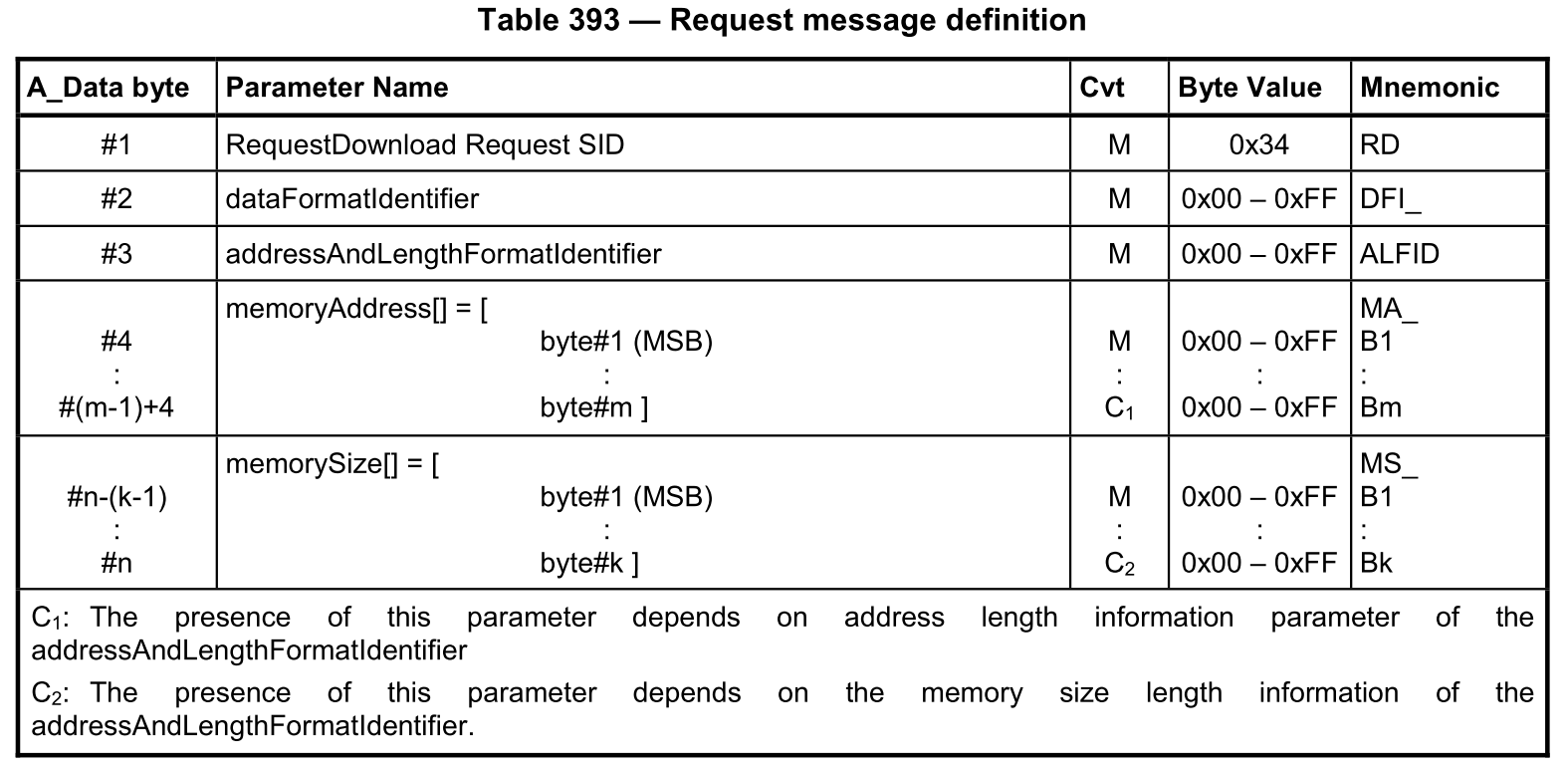
13.2.4 消极响应

|  |  |
| --- | --- |
| 0x12 | 子功能不支持 |
| 0x13 | 错误的消息长度或者无效的格式 |
| 0x22 | 条件不正确 |
| 0x24 | 请求序列错误  1 该例程当前处于活动状态，并且在收到“ startRoutine”子功能时无法重新启动（由汽车制造商决定是否可以在活动状态下重新启动给定的例程），   1. 当收到“ stopRoutine”子功能时，该例程当前未激活 2. 当收到'requestRoutineResults'子功能时，例行结果不可用（例如，从未启动所请求的例程Identifier）。 |
| 0x31 | 请求超出范围   1. routineIdentifier不支持 2. routineControlOptionRecord包含无效数据 |
| 0x33 | 安全访问拒绝 |
| 0x72 | 编程失败  有擦出和写入操作时，出现错误 |

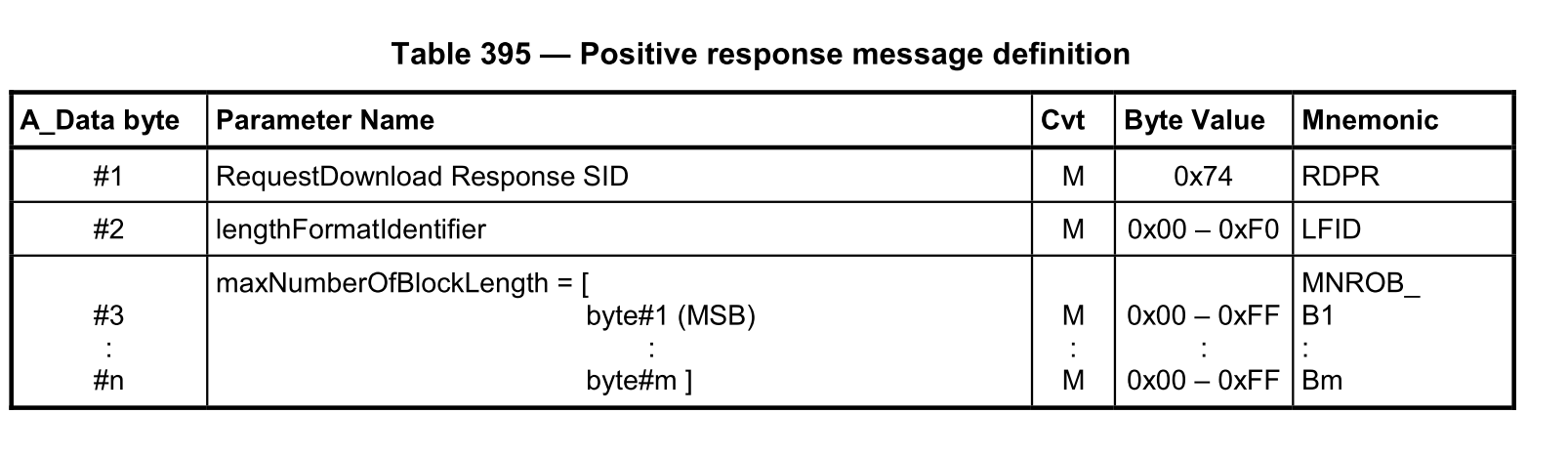


14.2 RequestDownload (0x34) service

用于ECU刷写或者标定。



|  |
| --- |
| dataFormatIdentifier:高4位指定了压缩方法，低4位指定了加密算法，0x00指定了既没有压缩方法也没有加密算法。 |
| addressAndLengthFormatIdentifier:  bit7-bit4: memorySize 所占字节数  bit3-bit0: memoryAddress 所占字节数 |
| memoryAddress:  参数memoryAddress是要写入数据的服务器内存的起始地址。用于此地址的字节数由addressAndLengthFormatIdentifier的低半字节（位3-0）定义。 memoryAddress参数中的字节#m始终是服务器中引用的地址的最低有效字节。地址的最高有效字节可以用作存储器标识符。  使用内存标识符的一个示例是具有16位地址和内存地址重叠的双处理器服务器（当给定地址对任一处理器均有效，但产生不同的物理内存设备或使用内部和外部闪存时）。在这种情况下，可以将memoryAddress参数中的其他未使用字节指定为用于选择所需存储设备的存储标识符。此功能的使用应由车辆制造商/系统供应商定义。 |
| memorySize:  服务器应使用此参数将内存大小与TransferData服务期间传输的数据总量进行比较。这增加了编程安全性。用于此大小的字节数由addressAndLengthFormatIdentifier的高半字节（位7-4）定义。如果使用数据压缩，则存储器大小是否代表压缩或未压缩大小取决于车辆制造商。 |



|  |
| --- |
| lengthFormatIdentifier:  bit7-4: maxNumberOfBlockLength所占字节数  bit3-0: 保留 |
| maxNumberOfBlockLength:  requestDownload肯定响应消息使用此参数来通知客户端来自客户端的每个TransferData请求消息中要包含多少个数据字节（maxNumberOfBlockLength）。该长度反映了完整的消息长度，包括服务标识符和TransferData请求消息中存在的数据参数。此参数允许客户端在开始将数据传输到服务器之前适应服务器的接收缓冲区大小。服务器需要接受与其报告的maxNumberOfBlockLength长度相等的transferData请求。特定于服务器的是什么长度小于maxNumberOfBlockLength的transferData请求被接受（如果有）。注意，给定块中的最后一个transferData请求可能需要小于maxNumberOfBlockLength。不允许服务器写入传输数据消息中未包含的其他数据字节（即填充字节）（压缩或未压缩格式），因为这会影响后续的传输数据请求数据所在的内存地址被写 |

14.2.4 消极响应

|  |  |
| --- | --- |
| 0x13 | 错误的消息长度或者无效的格式 |
| 0x22 | 条件不正确  如果服务器在接收软件或校准模块的下载过程中收到此服务的请求，则应返回此NRC。如果在下载模块期间服务器和客户端之间的数据大小不匹配，可能会发生这种情况。 |
| 0x31 | 请求超出范围   1. dataFormatIdentifier无效 2. addressAndLengthFormatIdentifier无效 3. memoryAddress/memorySize 无效 |
| 0x33 | 安全访问拒绝 |
| 0x70 | 上传下载不被接受 |

14.3 RequestUpload (0x35) service

14.4 TransferData (0x36) service

14.5 RequestTransferExit (0x37) service

14.6 RequestFileTransfer (0x38) service