|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | MAS 31 |
| Source:\* | Anupama, C-DOT, anupama@cdot.in Poornima, C-DOT, poornima@cdot.inGiri, C-DOT, moode@cdot.in |
| Date:\* | 2017-09-07 |
| Reason for Change/s:\* | See the introduction  |
| CR against: Release\* | Release 3 |
| CR against: WI\* | [ ]  Active <Work Item number> [x]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [ ]  No [x] mirror CR number: (Note to Rapporteur - use latest agreed revision)[ ]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0006 Version 3.5.0 |
| Clauses \* | 8.2 |
| Type of change: \* | [ ]  Editorial change[x]  Bug Fix or Correction[ ]  Change to existing feature or functionality[ ]  New feature or functionalityOnly ONE of the above shall be ticked |
| Impacted other TS/TR(s) | <TS/TR number>, <Version Number>, and <Description on which aspect should be reflected in this TS/TR> |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES [x]  NO [ ] This CR may break backwards compatibility with the last approved version of the TS? YES [ ]  NO [ ]  |
| Template Version: January 2017 (Do not modify) |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

In case of a correction, and the change apply to previous releases, a separate “mirror CR” should be posted at the same time of this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made E.g. A change impacting 5 tables should not only include a proposal to change only 3 tables. Includes any changes to references, definitions, and acronyms in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

Use Change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

## Introduction

The CR proposes to correct execStatus to execResult as it maps to execResult as per status given in TS-0004.

Table 6.3.4.2.15‑1: Interpretation of execStatusType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | INITIATED |  |
| 2 | PENDING |  |
| 3 | FINISHED |  |
| 4 | CANCELLING |  |
| 5 | CANCELLED |  |
| 6 | STATUS\_NON\_CANCELLABLE |  |
| NOTE: See clause 7.4.17 "Resource Type execInstance".  |

Table 6.3.4.2.16‑1: Interpretation of execResultType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 0 | STATUS\_SUCCESS |  |
| 1 | STATUS\_REQUEST\_UNSUPPORTED |  |
| 2 | STATUS\_REQUEST DENIED  |  |
| 3 | STATUS\_CANCELLATION\_DENIED |  |
| 4 | STATUS\_INTERNAL\_ERROR |  |
| 5 | STATUS\_INVALID\_ARGUMENTS |  |
| 6 | STATUS\_RESOURCES\_EXCEEDED |  |
| 7 | STATUS\_FILE\_TRANSFER\_FAILED |  |
| 8 | STATUS\_FILE\_TRANSFER\_SERVER\_AUTHENTICATION\_FAILURE |  |
| 9 | STATUS\_UNSUPPORTED\_PROTOCOL |  |
| 10 | STATUS\_UPLOAD\_FAILED |  |
| 11 | STATUS\_FILE\_TRANSFER\_FAILED\_MULTICAST\_GROUP\_UNABLE\_JOIN |  |
| 12 | STATUS\_FILE\_TRANSFER\_FAILED\_SERVER\_CONTACT\_FAILED |  |
| 13 | STATUS\_FILE\_TRANSFER\_FAILED\_FILE\_ACCESS\_FAILED |  |
| 14 | STATUS\_FILE\_TRANSFER\_FAILED\_DOWNLOAD\_INCOMPLETE |  |
| 15 | STATUS\_FILE\_TRANSFER\_FAILED\_FILE\_CORRUPTED |  |
| 16 | STATUS\_FILE\_TRANSFER\_FILE\_AUTHENTICATION\_FAILURE |  |
| 19 | STATUS\_FILE\_TRANSFER\_WINDOW\_EXCEEDED |  |
| 20 | STATUS\_INVALID\_UUID\_FORMAT |  |
| 21 | STATUS\_UNKNOWN\_EXECUTION\_ENVIRONMENT |  |
| 22 | STATUS\_DISABLED\_EXECUTION\_ENVIRONMENT |  |
| 23 | STATUS\_EXECUTION\_ENVIRONMENT\_MISMATCH |  |
| 24 | STATUS\_DUPLICATE\_DEPLOYMENT\_UNIT |  |
| 25 | STATUS\_SYSTEM\_RESOURCES\_EXCEEDED |  |
| 26 | STATUS\_UNKNOWN\_DEPLOYMENT\_UNIT |  |
| 27 | STATUS\_INVALID\_DEPLOYMENT\_UNIT\_STATE |  |
| 28 | STATUS\_INVALID\_DEPLOYMENT\_UNIT\_UPDATE\_DOWNGRADE\_DISALLOWED |  |
| 29 | STATUS\_INVALID\_DEPLOYMENT\_UNIT\_UPDATE\_UPGRADE\_DISALLOWED |  |
| 30 | STATUS\_INVALID\_DEPLOYMENT\_UNIT\_UPDATE\_VERSION\_EXISTS |  |
| 31 | STATUS\_NOT\_FINISHED |  |
| NOTE: See clause 7.4.16 "Resource Type mgmtCmd". |

Second change corrects status to “STATUS\_CANCELLATION\_DENIED” in clause 8.2.3

### -----------------------Start of change 1-------------------------------------------

## 8.2 <mgmtCmd> and <execInstance> resource primitive mappings

### 8.2.1 Update (Execute) primitive for the <mgmtCmd> resource

#### 8.2.1.0 Introduction

When the Update Request primitive for <mgmtCmd> resource addresses the execEnable attribute of the <mgmtCmd> resource, it effectively triggers an Execute <mgmtCmd> procedure.

The Hosting CSE performs command conversion of its <execInstance> sub-resources. The mapping between the <execInstance> attributes and the TR-069 [4] RPC procedures triggered is based on the value of the cmdType attribute of the <mgmtCmd> resource defined in Table 8.2.1.0-1. The CPE acceptance of the corresponding RPC procedures is indicated by returning a successful Response primitive to the initial Update Request.

The Fault Codes which may be returned by the CPE to the Hosting CSE are mapped onto execResult codes and stored in the corresponding <execInstance> attributes, and are detailed in the following clauses:

Table 8.2.1.0-1 Mapping of Execute <mgmtCmd> primitives to BBF TR-069 RPC

| cmdType value  | BBF TR-069 RPCs |
| --- | --- |
|  “DOWNLOAD” | Download RPC (see clause 8.2.1.1) and TransferComplete RPC (clause 8.2.1.3) |
| “UPLOAD” | Upload RPC (clause 8.2.1.2) and TransferComplete RPC (clause 8.2.1.3) |
| “SOFTWAREINSTALL” | ChangeDUState RPC (clause 8.2.1.4) and ChangeDUStateComplete RPC (clause 8.2.1.5) |
|  “SOFTWAREUNINSTALL” | ChangeDUState RPC (clause 8.2.1.4) and ChangeDUStateComplete RPC (clause 8.2.1.5) |
|  “REBOOT” | Reboot RPC (clause 8.2.1.6) |
|  “RESET” | Factory reset RPC (clause 8.2.1.7) |

#### 8.2.1.1 Execute File Download

The download file transfer operation may use the Download mechanism defined in TR-069 [4]. The Download mechanism is an asynchronous command which returns a successful response or one of the following fault codes mapped onto execResult values as detailed in Table 8.2.1.1-1. A successful response to the Update primitive triggering the Execute procedure means that the CPE has accepted the Download RPC.

Table 8.2.1.1-1: Download Fault Code Mapping

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9000 | Method not supported | STATUS\_REQUEST\_UNSUPPORTED |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST\_DENIED |
| 9002 | Internal error | STATUS\_INTERNAL\_ERROR |
| 9003 | Invalid arguments | STATUS\_INVALID\_ARGUMENTS |
| 9004 | Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error) | STATUS\_RESOURCES\_EXCEEDED |
| 9010 | File transfer failure (associated with Download, ScheduleDownload, TransferComplete or AutonomousTransferComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED |
| 9012 | File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods, not associated with Scheduled Download method). | STATUS\_FILE\_TRANSFER\_SERVER\_AUTHENTICATION\_FAILURE |
| 9013 | Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_UNSUPPORTED\_PROTOCOL |

#### 8.2.1.2 Execute File Upload Operations

The upload file transfer operation shall use the Upload mechanism defined in TR-069 [4]. The Upload mechanism is an asynchronous command that consists of the synchronous Upload RPC for the Upload and the asynchronous TransferComplete RPC. The Upload RPC returns a successful response or one of the following fault codes mapped onto execResult values as detailed in Table 8.2.1.2-1. A successful response to the Update primitive triggering the execute procedure means that the CPE has accepted the Upload RPC in Table 8.2.1.2-1.

Table 8.2.1.2-1: Upload Fault Code Mapping

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9000 | Method not supported | STATUS\_REQUEST\_UNSUPPORTED  |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST DENIED |
| 9002 | Internal error | STATUS\_INTERNAL\_ERROR  |
| 9003 | Invalid arguments | STATUS\_INVALID\_ARGUMENTS  |
| 9004 | Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error) | STATUS\_RESOURCES\_EXCEEDED  |
| 9011 | Upload failure (associated with Upload, TransferComplete or Autonomous­Trans­ferComplete methods). | STATUS\_UPLOAD\_FAILED  |
| 9012 | File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_SERVER\_AUTHENTICATION\_FAILURE |
| 9013 | Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_UNSUPPORTED\_PROTOCOL  |

#### 8.2.1.3 Report Results using TransferComplete RPC

After a File Download or Upload has been attempted, the result of the operation is reported using the TransferComplete RPC. The TransferComplete RPC indicates a successful operation or one of the following fault codes mapped onto execResult values in Table 8.2.1.3-2.

Table 8.2.1.3-2: TransferComplete Fault Code Mapping

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST DENIED  |
| 9002 | Internal error | STATUS\_INTERNAL\_ERROR  |
| 9010 | File transfer failure (associated with Download, ScheduleDownload, TransferComplete or AutonomousTransferComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED  |
| 9011 | Upload failure (associated with Upload, TransferComplete or Autonomous­Trans­ferComplete methods). | STATUS\_UPLOAD\_FAILED |
| 9012 | File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_SERVER\_AUTHENTICATION\_FAILURE |
| 9014 | File transfer failure: unable to join multicast group (associated with Download, TransferComplete or AutonomousTransferComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED\_MULTICAST\_GROUP\_UNABLE\_JOIN |
| 9015 | File transfer failure: unable to contact file server (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED\_SERVER\_CONTACT\_FAILED |
| 9016 | File transfer failure: unable to access file (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED\_FILE\_ACCESS\_FAILED  |
| 9017 | File transfer failure: unable to complete download (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED\_DOWNLOAD\_INCOMPLETE  |
| 9018 | File transfer failure: file corrupted or otherwise unusable (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED\_FILE\_CORRUPTED |
| 9019 | File transfer failure: file authentication failure (associated with Download, TransferComplete or AutonomousTransferComplete methods). | STATUS\_FILE\_TRANSFER\_FILE\_AUTHENTICATION\_FAILURE |
| 9020 | File transfer failure: unable to complete download within specified time windows (associated with TransferComplete method). | STATUS\_FILE\_TRANSFER\_WINDOW\_EXCEEDED |

#### 8.2.1.4 Execute Software Operations with ChangeDUState RPC

The software installation and uninstall operations shall use the ChangeDUState mechanism defined in TR-069 [4]. The ChangeDUState mechanism is an asynchronous command that consists of the synchronous ChangeDUState RPC and returns a successful response or one of the fault codes mapped onto execResult values as detailed in Table 8.2.1.4.-1. A successful response to the Update primitive triggering the Execute procedure means that the CPE has accepted the ChangeDUState RPC.

Table 8.2.1.4-1: ChangeDUState Fault Code Mapping

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9000 | Method not supported | STATUS\_REQUEST\_UNSUPPORTED |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST DENIED  |
| 9002 | Internal error | STATUS\_INTERNAL\_ERROR  |
| 9004 | Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error) | STATUS\_RESOURCES\_EXCEEDED  |

#### 8.2.1.5 Report Results with ChangeDUStateComplete RPC

After software installation and uninstall operations using a ChangeDUState mechanism as defined in TR-069 [4], the result of the state change operation is retrieved using the ChangeDUStateComplete RPC. The ChangeDUStateComplete RPC indicates a successful operation or one of the fault codes mapped onto execResult values as detailed in Table 8.2.1.5.-1.

Table 8.2.1.5-1: ChangeDUStateComplete Fault Code Mapping

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST\_DENIED  |
| 9003 | Invalid arguments | STATUS\_INVALID\_ARGUMENTS  |
| 9012 | File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_SERVER\_AUTHENTICATION\_FAILURE |
| 9013 | Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_UNSUPPORTED\_PROTOCOL  |
| 9015 | File transfer failure: unable to contact file server (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED\_SERVER\_CONTACT\_FAILED  |
| 9016 | File transfer failure: unable to access file (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED\_FILE\_ACCESS\_FAILED  |
| 9017 | File transfer failure: unable to complete download (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED\_DOWNLOAD\_INCOMPLETE  |
| 9018 | File transfer failure: file corrupted or otherwise unusable (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods). | STATUS\_FILE\_TRANSFER\_FAILED\_FILE\_CORRUPTED  |
| 9022 | Invalid UUID Format (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update, and Uninstall) | STATUS\_INVALID\_UUID\_FORMAT |
| 9023 | Unknown Execution Environment (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install only) | STATUS\_UNKNOWN\_EXECUTION\_ENVIRONMENT |
| 9024 | Disabled Execution Environment (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update, and Uninstall) | STATUS\_DISABLED\_EXECUTION\_ENVIRONMENT  |
| 9025 | Deployment Unit to Execution Environment Mismatch (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install and Update) | STATUS\_EXECUTION\_ENVIRONMENT\_MISMATCH  |
| 9026 | Duplicate Deployment Unit (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install only) | STATUS\_DUPLICATE\_DEPLOYMENT\_UNIT  |
| 9027 | System Resources Exceeded (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install and Update) |  STATUS\_SYSTEM\_RESOURCES\_EXCEEDED |
| 9028 | Unknown Deployment Unit (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update and Uninstall) | STATUS\_UNKNOWN\_DEPLOYMENT\_UNIT |
| 9029 | Invalid Deployment Unit State (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update and Uninstall) | STATUS\_INVALID\_DEPLOYMENT\_UNIT\_STATE |
| 9030 | Invalid Deployment Unit Update – Downgrade not permitted (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only) | STATUS\_INVALID\_DEPLOYMENT\_UNIT\_UPDATE\_DOWNGRADE\_DISALLOWED |
| 9031 | Invalid Deployment Unit Update – Version not specified (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only) | STATUS\_INVALID\_DEPLOYMENT\_UNIT\_UPDATE\_UPGRADE\_DISALLOWED |
| 9032 | Invalid Deployment Unit Update – Version already exists (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only) | STATUS\_INVALID\_DEPLOYMENT\_UNIT\_UPDATE\_VERSION\_EXISTS |

#### 8.2.1.6 Execute Reboot operation

The reboot operation shall use the Reboot RPC defined in TR-069 [4]. The Reboot RPC is a synchronous command. A successful response to the Update primitive triggering the Execute procedure means that the CPE has accepted the Reboot RPC. The Reboot RPC returns a successful response or one of the fault codes mapped onto execResult values as detailed in Table 8.2.1.6-1.

Table 8.2.1.6-1: Reboot Fault Code Mapping

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST\_DENIED  |
| 9002 | Internal error | STATUS\_INTERNAL\_ERROR  |
| 9003 | Invalid arguments | STATUS\_INVALID\_ARGUMENTS |

#### 8.2.1.7 Execute Factory Reset operation

The factory reset operation shall use the FactoryReset RPC defined in TR-069 [4]. The FactoryReset RPC is a synchronous command. A successful response to the Update primitive triggering the Execute procedure means that the CPE has accepted the FactoryReset RPC. The FactoryReset RPC returns a successful response or one of the fault codes mapped onto execResult values as detailed in Table 8.2.1.7-1.

Table 8.2.1.7-1: FactoryReset Fault Code Mapping

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9000 | Method not supported | STATUS\_REQUEST\_UNSUPPORTED  |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST\_DENIED  |
| 9002 | Internal error | STATUS\_INTERNAL\_ERROR |
| 9003 | Invalid arguments | STATUS\_INVALID\_ARGUMENTS  |

### 8.2.2 Delete <mgmtCmd> resource primitive mapping

The Delete Request primitive for the <mgmtCmd> resource may initiate TR-069 [4] RPC commands for the corresponding <execInstance> sub-resources as follows:

* If there are no <execInstance> sub-resources with RUNNING execStatus, a successful response to the Delete primitive is returned and the <mgmtCmd> resource is deleted without triggering any TR-069 [4] RPCs.
* If there are <execInstance> sub-resources with RUNNING execStatus that resulted in cancellable TR-069 [4] RPCs (e.g. File Upload and File Download RPCs), a TR-069 [4] CancelTransfer RPC shall be initiated for each cancellable operation. Upon completion of all the cancellation operations, if any fault codes are returned by the CPE, an unsuccessful Response to the Delete primitive with status code “Delete mgmtCmd- execInstance cancellation error” is returned, and the <mgmtCmd> resource is not deleted. The execStatus attribute of each specific <execInstance> is set to CANCELLED and the execResult attribute is set to “STATUS\_SUCCESS” for successful RPCs. For the unsuccessful case, execResult is determined from the RPC fault codes as detailed in Table 8.2.2-1. If all cancellation operations are successful on the managed entity, a successful Response to the Delete primitive is returned and the <mgmtCmd> resource is deleted.
* If there is at least one <execInstance> sub-resource with RUNNING execStatus that resulted in non-cancellable TR-069 [4] RPCs (e.g. RPCs other than File Upload and File Download RPCs), the execStatus attribute of the specific <execInstance> is changed to STATUS\_NON\_CANCELLABLE. An unsuccessful Response to the Delete primitive with status code “Delete mgmtCmd- execInstance cancellation error” is returned and the <mgmtCmd> resource is not deleted.

Table 8.2.2-1: CancelTransfer Fault Code Mapping for Delete <mgmtCmd>

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9000 | Method not supported | STATUS\_REQUEST\_UNSUPPORTED  |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST DENIED  |
| 9021 | Cancellation of file transfer not permitted in current transfer state | STATUS\_CANCELLATION\_DENIED  |

### 8.2.3 Update (Cancel) <execInstance> primitive mapping

When the Update Request primitive for an <execInstance> sub-resource addresses the execDisable attribute of the <execInstance > sub-resource, it effectively triggers a Cancel <execInstance> resource procedure.

The hosting CSE determines whether the <execInstance> resource has a RUNNING execStatus and weather the resulting TR-069 [4] RPCs are cancellable. Currently, only the TR-069 File Upload and File Download RPCs are cancellable using the TR-069 [4] CancelTransfer RPC.

* If the addressed <execInstance> sub-resource has an execStatus other than RUNNING, an un-successful Response to the Update primitive is returned with status code “Cancel execInstance – already complete”.
* If the addressed <execInstance> sub-resources has RUNNING execStatus and resulted in cancellable TR-069 [4] RPCs (e.g. File Upload and File Download RPCs), a BBF TR-069 [4] CancelTransfer RPC shall be initiated. For a successful CancelTransfer RPC the execStatus attribute of the specific <execInstance> is set to CANCELLED, the execResult attribute is set to “STATUS\_SUCCESS” and a successful Response is sent to the Update primitive. For an unsuccessful CancelTransfer RPC the execResult attribute is determined from the RPC fault codes as detailed in Table 8.2.3-1 and an unsuccessful Response is sent to the Update primitive with status code “Cancel execInstance – cancellation error”.
* If the addressed <execInstance> sub-resources has RUNNING execStatus and resulted non-cancellable TR-069 [4] RPCs (e.g. RPCs other than File Upload and File Download RPCs), the execStatus attribute of the specific <execInstance> is changed to STATUS\_NON\_CANCELLABLE. An unsuccessful Response is sent to the Update primitive with status code “Cancel execInstance – not cancellable”.

Table 8.2.3-1: CancelTransfer Fault Code Mapping for Update (Cancel) <execInstance>

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9000 | Method not supported | STATUS\_REQUEST\_UNSUPPORTED  |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST DENIED  |
| 9021 | Cancellation of file transfer not permitted in current transfer state | STATUS\_CANCELLATION\_DENIED |

### 8.2.4 Delete <execInstance> primitive mapping

The Delete Request primitive for an <execInstance> sub-resource may initiate TR-069 [4] RPC commands for the corresponding <execInstance> sub-resources as follows:

* If the addressed <execInstance> sub-resource has an execStatus other than RUNNING, an successful Response to the Delete primitive is returned and the <execInstance> sub-resource is deleted without triggering any TR-069 [4] RPCs.
* If the addressed <execInstance> sub-resource has RUNNING execStatus and resulted in cancellable TR-069 [4] RPCs (e.g. File Upload and File Download RPCs), a BBF TR-069 [4] CancelTransfer RPC shall be initiated. For a successful CancelTransfer RPC a successful response is sent to the Delete primitive and the <execInstance> sub-resource is deleted. For an unsuccessful CancelTransfer RPC the execResult attribute is determined from the RPC fault codes as detailed in Table 8.2.4-1 and an unsuccessful Response is sent to the Delete primitive with status code “Delete execInstance – cancellation failed”.
* If the addressed <execInstance> sub-resource has RUNNING execStatus and resulted non-cancellable TR-069 [4] RPCs (e.g. RPCs other than File Upload and File Download RPCs), the execStatus attribute is set to STATUS\_NON\_CANCELLABLE and an unsuccessful Response is sent to the Update primitive with status code “Delete execInstance – not cancellable”

Table 8.2.4-1: CancelTransfer Fault Code Mapping for Delete <execInstance>

| Fault code | Description | execResult Code |
| --- | --- | --- |
| 9000 | Method not supported | STATUS\_REQUEST\_UNSUPPORTED  |
| 9001 | Request denied (no reason specified) | STATUS\_REQUEST DENIED  |
| 9021 | Cancellation of file transfer not permitted in current transfer state | STATUS\_CANCELLATION\_DENIED  |

### -----------------------End of change 1---------------------------------------------

CHECK LIST

* Does this Change Request include an informative introduction containing the problem(s) being solved, and a summary list of proposals.?
* Does this CR contain changes related to only one particular issue/problem?
* Have any mirror CRs been posted?
* Does this Change Request make **all** the changes necessary to address the issue or problem? E.g. A change impacting 5 tables should not include a proposal to change only 3 tables?Does this Change Request follow the drafting rules?
* Are all pictures editable?
* Have you checked the spelling and grammar?
* Have you used change bars for all modifications?
* Does the change include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change? (Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.)
* Are multiple changes in this CR clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.?