

“WEB SERVICES”

Sync Web Services

url:

<https://agilege4reporting.us.dell.com:7002/AgileSyncSWService/services/AgileSyncSWService>

wsdl:

<https://agilege4reporting.us.dell.com:7002/AgileSyncSWService/services/AgileSyncSWService?wsdl>

Credentials:

- Username: Americas\serviceagileprod
- Password: *****

cancelGDEV

➤ **Method Description:**

- The method cancelGDEV is part of a service that cancels a GDEV Deviation in an Agile PLM system. The method involves several steps, including session creation, validation, and status change, and is equipped with logging and error handling.

➤ **Working:**

- The method cancelGDEV takes a GDEVCancelRequest object as input and returns a GDEVCancelResponse object.
- GDEVCancelResponse object is created to store the response, variables gdevNumber, session, and gDev are initialized.
- Properties are loaded from the specified configuration files.
- Session Creation: Creates an Agile session using decrypted credentials.
- GDEV Retrieval and Validation: Retrieves the GDEV object and validates its status.
- Status Change: Changes the GDEV status to "Canceled" if it meets the criteria.
- Error Handling: Provides detailed error messages for validation and other exceptions.
- Resource Management: Ensures the Agile session is properly closed to free resources.

- Parameters:
 - FileFolderRequest fileReq
- Response:
 - Response will content GDEVCancelResponse object.

createAutoSWBPart

- Method Description:
 - The method createAutoSWBPart is designed to automate the creation of a new SWB in an Agile PLM system based on a request.
 - The method handles session creation, validation, and the actual creation of the SWB, with extensive logging and error handling.
- Working:
 - createAutoSWBPart takes a SWBCreateRequest object as input and returns a SWBAutoCreationResponse object.
 - A SWBAutoCreationResponse object is created to store the response,
 - Logging and Property Loading: Logs important steps and loads necessary properties.
 - Session Creation: Creates an Agile session using decrypted credentials.
 - User Validation: Validates the current user against the session.
 - SWB Creation: Creates either an independent or successor SWB based on the request.
 - Response Population: Sets the response status, new SWB number, and error messages.
 - Error Handling: Provides detailed error messages for validation and other exceptions.
 - Resource Management: Ensures the Agile session is properly closed to free resources.
- Parameters:
 - SWBCreateRequest swbCreateRequest

- Response:
 - Response will content SWBAutoCreationResponse object.

createPart

- Method Description:
 - The createPart method is designed to create a new part in the Agile PLM system using the details provided in the PartCreateRequest object. It involves setting up the session, validating inputs, creating the part, setting attributes, and handling various exceptions.
- Working:
 - createPart takes a PartCreateRequest object as input and returns a CreatePartResponse object.
 - Variable session, admin, cls, numSources, object, partClass, and partCategory, are initialized.
 - Logging and Property Loading: Logs the process and loads necessary properties.
 - Session Creation: Creates an Agile session using provided credentials.
 - Part Creation: Uses auto-numbering to create a new part with the specified class and category.
 - Attribute Setting: Sets additional attributes on the created part if provided.
 - Error Handling: Handles and logs different types of exceptions, setting appropriate error messages in the response.
 - Session Management: Ensures the Agile session is properly closed to free resources.
- Parameters:
 - PartCreateRequest partCreateRequest
- Response:
 - Response will content CreatePartResponse object.

getDpkPartAttribute

➤ Method Description:

- This getDpkPartAttribute method is a method to retrieve attributes of a part given its part number.

➤ Working:

- This getDpkPartAttribute takes a String parameter partNumber and returns a PartResponse object.
- A PartResponse object named response is initialized to store the part attributes.
- Logging: Logs the start of the method execution.
- Part Retrieval: Uses DpkPartretriveBO to retrieve and populate part attributes into a PartResponse object.
- Exception Handling: Catches and logs any exceptions.
- Logging: Logs the end of the method execution.
- Return: Returns the populated PartResponse object.

➤ Parameters:

- String partNumber

➤ Response:

- Response will content PartResponse object.

➤ Example:

- Request from Downstream



- Response from Agile



```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:plm="http://agilesyncswservice.integration.agile.plm">
  <soapenv:Body>
    <getDpkPartAttributeResponse xmlns="http://agilesyncswservice.integration.agile.plm">
      <getDpkPartAttributeReturn>
        <dbFailure xsi:nil="true"/>
        <objectId>01WC9</objectId>
        <partAttributeXML><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
          <PartList>
            <Row>
              <Column name="Part Class">Digital Product Key</Column>
              <Column name="Description">DPK, SWV, LKEY, HMNG, AP, PERP, SYR</Column>
              <Column name="License Binding Type">Trusted</Column>
              <Column name="Part Number">01WC9</Column>
              <Column name="License is an upgrade">No</Column>
              <Column name="KeyGenPartNumber">No</Column>
              <Column name="License Duration"/>
              <Column name="License is transferable">YES</Column>
              <Column name="License Type">Perpetual</Column>
              <Column name="Part Type">DPK, SW Virtual</Column>
            </Row>
          </PartList>
        ]]></partAttributeXML>
        <status>Part Details have been successfully retrieved</status>
        <validationError/>
      </getDpkPartAttributeReturn>
    </getDpkPartAttributeResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

getFileFolderDetails

➤ Method Description:

- The getFileFolderDetails method retrieves details about files attached to a specific part in Agile PLM based on a given request. It returns an array of FileFolderResponse objects containing the file details.

➤ Working:

- The method takes a FileFolderRequest object fileReq as input and returns an array of FileFolderResponse objects.
- Variable are initialized.
- Logging: Logs the start of the method execution.
- Properties Loading: Loads configuration properties.
- Request Parameter Extraction: Extracts and validates input parameters.

- Agile Session Creation: Creates a session with Agile PLM.
- Part Existence Check: Validates the existence of the specified part.
- Attachments Retrieval: Retrieves file details attached to the part.
- Response Construction: Constructs response objects with file details.
- Error Handling: Handles exceptions and logs errors.
- Session Cleanup: Ensures the Agile session is closed.
- Return Response: Returns an array of FileFolderResponse objects.

➤ Parameters:

- FileFolderRequest fileReq

➤ Response:

- Response will content FileFolderResponse object.

getOSLangMatrix

➤ Method Description:

- This The getOSLangMatrix method is designed to retrieve the OS Language Matrix for a given SWB part number.

➤ Working:

- The method takes a String parameter swbPartNumber and returns a Response object.
- Variable are initialized.
- Logging Configuration: Configures Log4J for logging.
- Input Validation: Checks if swbPartNumber is provided and appends an error message if not.
- Properties Loading: Logs the start of property loading (although actual loading is not shown in the snippet).
- Validation Check: If validation errors are found, logs the error and throws a ValidationException.
- OS Language Matrix Retrieval: If no validation errors, creates an instance of OsLangMatrixBO and calls its getOsLangXMLResponse method to populate the response.

- Exception Handling: Catches ValidationException, logs the error, sets the response status to "Failure", and sets the validation error message.
- Return Response: Returns the populated Response object.

➤ Parameters:

- String swbPartNumber

➤ Response:

- Response will contain Response object.

➤ Example:

- Request from Downstream



```

Request 1
https://aqilereporting.us.dell.com:7002/AqileSyncSWService/services/AqileSyncS
XML
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:agil="http://agilesyncswservice.integration.agile.pl">
  <soapenv:Header/>
  <soapenv:Body>
    <agil:getOSLangMatrix>
      <agil:swbPartNumber>1YD58</agil:swbPartNumber>
    </agil:getOSLangMatrix>
  </soapenv:Body>
</soapenv:Envelope>

```

- Response from Agile



```

WService
XML
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:x="http://agilesyncswservice.integration.agile.pl">
  <soapenv:Body>
    <getOSLangMatrixResponse xmlns="http://agilesyncswservice.integration.agile.pl">
      <getOSLangMatrixReturn>
        <dbFailure xsi:nil="true"/>
        <objectId>1YD58</objectId>
        <osFileTypeCodeXML xsi:nil="true"/>
        <osLangXML><![CDATA[<?xml version="1.0" encoding="UTF-8"?><osl:OS-LANG
        <osl:os os-name="Microsoft - Windows Server 2012 - Datacenter # WS12D">
          <osl:language>English</osl:language>
        </osl:languages>
        </osl:os>
        </osl:OS-LANG>]]></osLangXML>
        <preReqCoreqXML xsi:nil="true"/>
        <status>Success</status>
        <swbPredecessorXML xsi:nil="true"/>
        <validationError/>
      </getOSLangMatrixReturn>
    </getOSLangMatrixResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

getOsToFileTypeCodeMapping

➤ Method Description:

- The getOsToFileTypeCodeMapping method is designed to retrieve the mapping between operating systems and file type codes.

➤ Working:

- Logging Configuration: Configures Log4J for logging.
- Method Start Logging: Logs the start of the getOsToFileTypeCodeMapping method.
- BO Instance Creation: Creates an instance of OsToFileTypeCodeBO.
- Calls the getOsToFileTypeCodeXMLResponse method on the OsToFileTypeCodeBO instance, which populates the response object with the OS to file type code mapping.
- Exception Handling: Catches any exceptions, logs the error, sets the response status to "Failure", and sets the validation error message.
- Method End Logging: Logs the end of the method.
- Return Response: Returns the populated Response object.

➤ Parameters:

- No Parameters

➤ Response:

- Response will content Response Object

➤ Example:

- Request from Downstream



- Response from Agile



getPhByPartNumber

➤ Method Description:

- This method takes PartNumberRequest class's object as a parameter which is in file PartNumberRequest.java. PartNumberRequest contains one class variable as String partNumber.

➤ Working:

- This method itself contains the code to connect with DB for fetching the data. The method is following the procedure mentioned under "DB Connection" section. But, the getDBConnection() method from DBConnectionManagerSingleton class is used instead of getConnection() method from "DB Connection" section.
- After getting the DB Connection, the method is calling the prepareCall() method on the object aglServiceconnection of class Connection.
- The prepareCall() method is getting the "{call SP_GET_ALL_PHS_FOR_ITM(?,?)}" String as parameter. The value

returned by the `prepareCall()` method is stored in the `CallableStatement` class's `cstmt` object.

- The method is then calling the `setString()` and `registerOutParameter()` method on the `cstmt` object as follows :
 - `cstmt.setString(1, partNumberRequest.getPartNumber());`
 - `cstmt.registerOutParameter(2, OracleTypes.CURSOR);`
 - Then the `executeUpdate()` method is being called on the `cstmt` object.
- The method is then adding the ph numbers in the response object and it is returned.

➤ Parameters:

- PartNumberRequest partNumberRequest

➤ Response:

- Response will content PartPhResponse object.

➤ Example:

- Request from Downstream



- Response from Agile



getPredecessorsForSwbNumber

➤ Method Description:

- The getPredecessorsForSwbNumber method process the swbNumber and calls the getPredecessorXMLResponse method and returns Response object.

➤ Working:

- Logging Configuration: Configures logging and logs the start of the method.
- Input Validation: Validates the swbNumber input and appends an error message if it's empty.
- Validation Check: If validation errors exist, logs the error, throws a ValidationException, and updates the response with failure status and error message.
- If validation passes, it calls the business logic method to get the predecessors.
- Business Logic: Uses an instance of SWBPredecessorSuccessorBO to call the getPredecessorXMLResponse method, passing the swbNumber and response object.
- Exception Handling: Catches ValidationException, logs the error, and updates the response with failure status and error message.
- Logging and Return: Logs the end of the method and returns the Response object.

➤ Parameters:

- String swbNumber

➤ Response:

- Response will content Response object.

getPrereqCoreqDetails

➤ Method Description:

- The getPrereqCoreqDetails method is designed to fetch prerequisite and corequisite details for a given software bundle number

(swbNumber). It returns a Response object that contains these details or error messages if any issues arise during the process.

➤ Working:

- Logging Configuration: Configures logging and logs the start of the method.
- Input Validation: Validates the swbNumber input and logs a message if it is empty.
- Validation Check: If validation errors exist, constructs an error message, logs it, throws a ValidationException, and updates the response with failure status and error message.
- If validation passes, it calls the business logic method to get the prerequisite and corequisite details.
- Business Logic: Uses an instance of SWBPrereqCoreqBO to call the getPrereqCoreqXMLResponse method, passing the swbNumber and response object.
- Exception Handling: Catches ValidationException, logs the error, and updates the response with failure status and error message.
- Logging and Return: Logs the end of the method and returns the Response object.

➤ Parameters:

- String swbNumber

➤ Response:

- Response will contain Response object.

➤ Example:

- Request from Downstream



- Response from Agile



getSWBDetails

➤ Method Description:

- This method getSWBDetails retrieves details related to software bundles based on the provided part number, performs necessary input validation, and handles exceptions.

➤ Working:

- Input Validation: Checks if the part number is empty or invalid.
- Load Configuration: Loads application and database properties.
- Create Agile Session: Establishes a session with Agile using credentials from the properties file.
- Retrieve Item: Retrieves the item object based on the part number.
- Validates if the item is a software bundle by checking its part class and part type.
- Error Handling: If validation errors occur, logs the error, throws a ValidationException, and updates the response with error details.
- Retrieve Part Details: If no validation errors, calls business logic to get part details.
- Cleanup: Closes the Agile session and nullifies objects to free resources.

- Return Response: Returns an SWBResponse object containing the part details or error messages.

➤ Parameters:

- String partNumber

➤ Response:

- Response will content SWBResponse object.

➤ Example:

- Request from Downstream



The screenshot shows a SOAP client interface with a toolbar and a URL bar containing the endpoint: `https://aqilereporting.us.dell.com:7002/AqileSyncSWService/services/AqileSyncSW`. The XML view shows the following request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:agil="h"
  <soapenv:Header/>
  <soapenv:Body>
    <agil:getSWBDetails>
      <agil:partNumber>053MDR</agil:partNumber>
    </agil:getSWBDetails>
  </soapenv:Body>
</soapenv:Envelope>
```

- Response from Agile



The screenshot shows the same SOAP client interface displaying the response XML. The XML structure is as follows:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:
  <soapenv:Body>
    <getSWBDetailsResponse xmlns="http://agilesyncswservice.integration.agile.plm
      <getSWBDetailsReturn>
        <SWBDetailsXML><![CDATA[<?xml version="1.0" encoding="UTF-8"?><Part:SW
          <swb:predecessors>
            <swb:predecessor/>
          </swb:predecessors>
        </Part:SWB-Details>]]></SWBDetailsXML>
        <dbFailure xsi:nil="true"/>
        <objectId>053MDR</objectId>
        <status>Success</status>
        <validationError/>
      </getSWBDetailsReturn>
    </getSWBDetailsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

getSWBFirstLevelWhereUsed

➤ Method Description:

- The getSWBFirstLevelWhereUsed method retrieves first-level parent part numbers for a given SWB number, ensuring proper validation and session management. It handles exceptions gracefully and logs detailed information throughout the process. The method returns a response object containing the status, SWB number, message ID, and parent part numbers.

➤ Working:

- The method takes an SWBFirstLevelWURequest object as a parameter and returns an SWBFirstLevelWUResponse object.
- An instance of SWBFirstLevelWUResponse is created.
- Various variables such as whereused, results, itr, row, parentPartNumbers, and session are initialized.
- The method starts by configuring Log4J for logging purposes.
- Loading Properties: The method retrieves the configuration base path and loads application properties from the aic_syncswservice_application.properties file.
- Standard system configuration properties are loaded from a specified path.
- Input Validation: The SWB number is retrieved from the request object and validated using the Utility.validateMandatoryAttributes method.
- Attributes Messageld and DownStreamName are validated.
- Agile session is created using the AgileConnect class, which retrieves session information from Node URLs.
- SWB Validation: The method checks if the given SWB number is valid using the SWBAutoGeneration.isValidSWB method. If invalid, a ValidationException is thrown.
- Retrieve Parent Part Numbers: The method uses the Utility.getWhereUsedFromTable method to retrieve parent part numbers for the given SWB number and appends them to the parentPartNumbers StringBuffer.
- If parent part numbers are found, the last character (comma) is removed.

- The response object is populated with the retrieved data, including the SWB number, message ID, and parent part numbers.
- If a ValidationException occurs, the response status is set to "Failure," and the error message is included.
- Other exceptions are caught and logged, and the response status is similarly set to "Failure."
- The Agile session is closed and set to null in the finally block to ensure proper resource management.

➤ Parameters:

- SWBFirstLevelWURequest swbFirstLevelWURequest

➤ Response:

- Response will content SWBFirstLevelWUResponse object.

➤ Example:

- Request from Downstream



- Response from Agile



getSWBStatus

➤ Method Description:

- The getSWBStatus method validates a given part number and determines the status of the associated Software Bundle (SWB). It handles input validation, retrieves status information, and manages exceptions gracefully. The method returns an AFPSResponse object containing the status, object ID, and any validation errors if applicable.

➤ Working:

- Initialize response and configure Log4J.
- Load application properties.
- Validate the input part number.
- Retrieve SWB status using SWBStatusBO.
- Handle exceptions and log information.
- Return the response object with status information.

➤ Parameters:

- String partNumber

➤ Response:

- Response will content AFPSResponse object.

getSuccessorsForSwbNumber

➤ Method Description:

- The getSuccessorsForSwbNumber method is designed to fetch details for a given swbNumber software bundle number.

➤ Working:

- The method takes a String parameter swbNumber and returns a SuccessorResponse object.
- Initializes a SuccessorResponse object and a StringBuffer for collecting error messages.
- Configures logging and logs the start of the method.
- Validates the swbNumber input and logs a message if it is empty.
- If validation errors exist, constructs an error message, logs it, throws a ValidationException, and updates the response with failure status and error message.
- If validation passes, it calls the business logic method to get the successor details.
- Uses an instance of SWBPredecessorSuccessorBO to call the getSuccessorXMLResponse method, passing the swbNumber and response object.
- Catches ValidationException, logs the error, and updates the response with failure status and error message.
- Logs the end of the method and returns the SuccessorResponse object.

➤ Parameters:

- String swbNumber

➤ Response:

- Response will contain SuccessorResponse object.

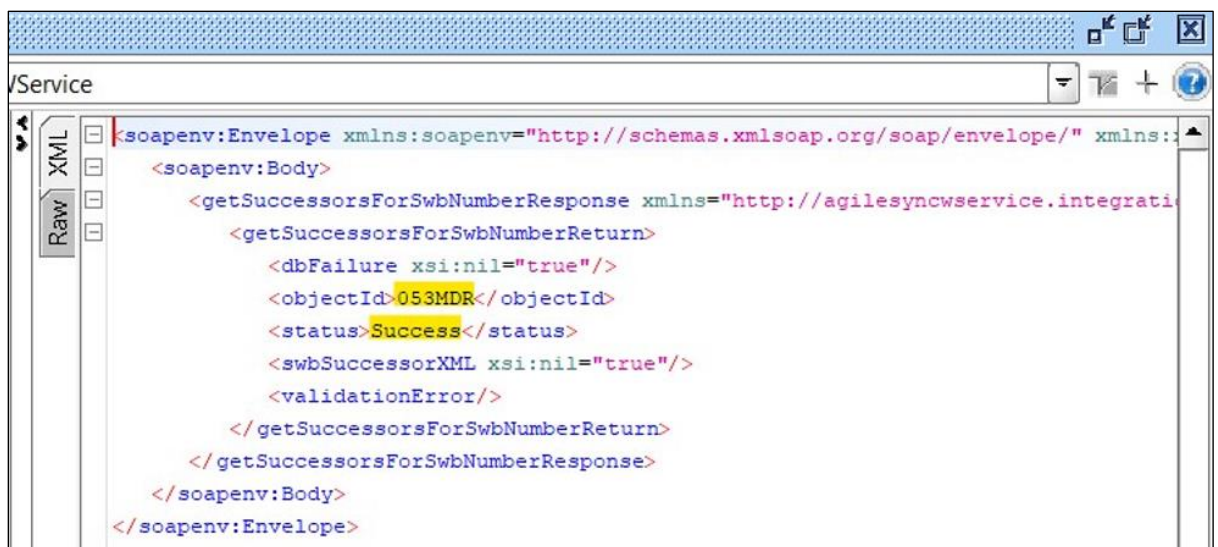
➤ Example:

- Request from Downstream



```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:agil="http://agilesyncswservice.integrations.dell.com">
  <soapenv:Header/>
  <soapenv:Body>
    <agil:getSuccessorsForSwbNumber>
      <agil:swbNumber>053MDR</agil:swbNumber>
    </agil:getSuccessorsForSwbNumber>
  </soapenv:Body>
</soapenv:Envelope>
```

- Response from Agile



```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:agil="http://agilesyncswservice.integrations.dell.com">
  <soapenv:Body>
    <getSuccessorsForSwbNumberResponse xmlns="http://agilesyncswservice.integrations.dell.com">
      <getSuccessorsForSwbNumberReturn>
        <dbFailure xsi:nil="true"/>
        <objectId>053MDR</objectId>
        <status>Success</status>
        <swbSuccessorXML xsi:nil="true"/>
        <validationError/>
      </getSuccessorsForSwbNumberReturn>
    </getSuccessorsForSwbNumberResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

swbAttributesUpdate

- Method Description:

- The swbAttributesUpdate method updates the attributes of a Software Bundle (SWB) based on the provided request. It performs several steps including session management, validation, attribute updating, and exception handling. The method returns a SWBAttributesUpdateResponse object containing the status of the update operation, including any error messages if applicable.

- Working:

- Initialize response and configure Log4J.
- Load application properties and validate mandatory attributes.

- Create Agile session and load SWB item.
- Validate SWB and update attributes if valid.
- If the session and SWB item are valid, the method validates the SWB attributes.
- If the SWB is valid, the method proceeds to validate conditions and update attributes.
- If attribute details are provided in the request, they are validated and updated using the `SwbAttributeUpdateValidations.updateAttributes` method.
- A new PNCR number is generated if attributes are successfully updated.
- Handle exceptions and log information.
- Close session and return response object with status and messages.

➤ Parameters:

- `SWBAttributesUpdateRequest swbAttributeUpdteRequest`

➤ Response:

- Response will content `SWBAttributesUpdateResponse` object.

updateAFPNumbers

➤ Method Description:

- The `updateAFPNumbers` method updates AFP numbers for a given SWB number by performing several validations checks and operations. It manages session creation and closure, validates input parameters and user credentials, removes invalid AFP and SRV numbers, updates AFP values, and logs all activities. The method ensures that exceptions are handled gracefully and that the Agile session is properly closed, returning an `AFPResponse` object indicating the status of the update operation.

➤ Working:

- The method starts by logging the beginning of the `updateAFPNumbers` method and `AFPResponse` object is initialized to store the response.

- Variables for the Agile session (IAgileSession) and item (IItem) are initialized to null.
- The method loads application properties from a configuration file using the FilePropertyUtility.loadProperties method.
- An AFPUUpdateBO object is created to handle business logic operations related to AFP updates.
- The service account is validated using the checkServiceAccValid method of the AFPUUpdateBO class.
- The method validates the input parameters (swbNumber, AFPSRVMap, userId) using the checkInputParamMandate method of the AFPUUpdateBO class.
- An Agile session is created using the createAgileSession method of the AFPUUpdateBO class.
- The user is validated within the Agile session using the checkUserValid method of the AFPUUpdateBO class.
- The SWB item is retrieved from the Agile session using the getObject method.
- The SWB number is validated using the checkSwbValid method of the AFPUUpdateBO class.
- Invalid AFP numbers are removed using the removeInvalidAfpNums method.
- Invalid SRV values are removed using the removeInvalidSrvValues method.
- AFP values are updated using the updateAfpValues method of the AFPUUpdateBO class.
- The history table is updated with the new values using the updateHistoryTable method of the AFPUUpdateBO class.
- If the update is successful, the response status is set to "Success".
- If any validation or API exceptions occur, the response status is set to "Failure" and the error message is logged and included in the response.
- The method handles various exceptions (ValidationException, APIException, and general Exception) and sets the appropriate error messages in the response.
- The Agile session is closed in the finally block to ensure proper resource management.

- The method logs the end of execution and returns the response object.

➤ Parameters:

- String swbNumber
- HashMap<String,String> AFPSRVMap
- String userId

➤ Response:

- Response will content AFPSRVResponse object.

updateGdevAttributes

➤ Method Description:

- The updateGdevAttributes method updates the "Effective To Date" attribute for a given Gdev number by performing several validation checks and operations. It manages session creation and closure, validates input parameters, checks the date format, updates the attribute, and logs all activities. The method ensures that exceptions are handled gracefully and that the Agile session is properly closed, returning an UpdateEffectiveToDateResponse object indicating the status of the update operation.

➤ Working:

- Configures logging and loads necessary properties.
- An UpdateEffectiveToDateResponse object is initialized to store the response.
- Variables for the Gdev number (String), Agile session (IAgileSession), and Gdev item (IChange) are initialized to null.
- The method loads application properties from a configuration file using the FilePropertyUtility.loadProperties method.
- The method retrieves the Gdev number and "Effective To Date" value from the request object.
- It validates the mandatory attributes ("Effective To Date" and "Global Deviation Number") using the Utility.validateMandatoryAttributes method.

- An Agile session is created using the `AgileConnect.createAgileSession` method.
- The Gdev item is retrieved from the Agile session using the `getObject` method.
- If the Gdev item is null (i.e., invalid Gdev number), the method logs an error and sets the response status to "Failure".
- The method creates a `GlobalDeviationAttributeUpdateBo` object to handle business logic operations related to Gdev attribute updates.
- It checks if the "Effective To Date" value is in the correct format ("MM/dd/yyyy") using the `isValidFormat` method.
- If the date format is invalid, the method logs an error and sets the response status to "Failure".
- If the date format is valid, the method updates the "Effective To Date" attribute using the `updateEffectToDate` method of the `GlobalDeviationAttributeUpdateBo` class.
- If the update is successful, the response status is set to "Success".
- If the update fails, the response status is set to "Failure" with the appropriate error message.
- The method handles various exceptions (`ValidationException`, `FileNotFoundException`, `IOException`, `APIException`, and `general Exception`) and sets the appropriate error messages in the response.
- The Agile session is closed in the finally block to ensure proper resource management.
- The session and Gdev number variables are set to null.
- The method logs the end of execution and returns the response object.

➤ Parameters:

- `UpdateEffectiveToDateRequest` `updateEffectToRequest`

➤ Response:

- Response will content `UpdateEffectiveToDateResponse` object.

updatePartAttribute

➤ Method Description:

- The `updatePartAttribute` method updates a specific attribute of a part identified by a given Software Bundle Number in an Agile PLM system. It performs various validation checks on the input parameters, manages Agile session creation and closure, and updates the attribute if all validations pass. The method ensures that exceptions are handled gracefully and logs all activities for debugging purposes. It returns a `PartResponse` object indicating the status of the update operation.

➤ Working:

- Configures logging and initializes local variables.
- Creates an Agile session using credentials and URL retrieved from properties.
- Checks if the current user is allowed to update the attribute.
- If not allowed, sets the response status to "Failure" and logs an error.
- Validates the SWB number, revision, attribute name, and attribute value.
- Retrieves the part item from the Agile session.
- Ensures the part item and attribute exist and are correct.
- If validations pass, updates the part item's revision.
- Checks if the destination is "Factory Install".
- If valid, updates the attribute using `UpdatePartAttributeBO`.
- Sets the response status based on the update's success or failure.
- Sets the response status to "Failure" and includes error messages.
- Closes the Agile session in a finally block to ensure proper resource management.
- Returns the `PartResponse` object indicating the success or failure of the update operation.

➤ Parameters:

- `String swbNumber`
- `String revision`
- `String attributeName`
- `String attributeValue`

➤ Response:

- Response will contain `PartResponse` object.

Note:

The below mentioned are some of few downstream names which are validated in the methods of some SOAP APIs:

- Digital Fulfillment
- Glovia
- SCDH(EBM)
- E2open (ITEM)
- PESO
- SPM (SPM)
- SDR
- Client Product Group (CPG)
- ESupport (ESP)
- Global Trade Management (GTM)