

1 TransUC3 XML-Interface

- ☐ with the latest TransUC3 version 2.1.0.0 it's possible to handle all new profile based configurations which start with UC3 application version 1.12.0
- ☐ for the former used commands is a possibility to backwards compatible by changing the value of a new customer.ini parameter

1.1 New TransUC3 commands to handle profile based configurations

- ☐ starting with UC3 application version 1.12.0 and TransUC3 version 2.1.0.0 all profile based configurations can be handled via XML-Interface with the new TransUC3 commands:
 - configuration profiles ⇒ NSM items 811 till 815
 - independent configuration profiles ⇒ NSM items 81+1 till 81+5
 - mode profiles ⇒ NSM items 821 till 825
 - device profiles\profile assignments ⇒ NSM item 831
 - device profiles ⇒ NSM items 832 till 836
- ☐ additional there are two new commands for
 - scale network functions ⇒ NSM item 84
 - software options ⇒ not available in the NSM
- ☐ more details about and how to use this commands you can find in the TransUC3 documentation "External TransUC3 documentation about the XML command interface for UC3 devices" which is part from the TransUC3 program folder in the subfolder \Doc

1.2 Backward compatibility with old xml commands managing application configurations

- ☐ there's a request to keep old XML configuration commands working which are
 - sales_application_configuration
 - selfservice_application_configuration
 - inventory_application_configuration
 - prepack_application_configuration
 - cashregister_application_configuration
- ☐ these listed former used XML commands have been defined for single devices without any profile usage
- ☐ to be backwards compatible this commands must be recognized and interpreted in a way, that they finally refer to profile based configurations which are assigned to the device number referenced in the old command (the device where command is received and interpreted)
- ☐ these old device number based commands use pairs of configuration property names and configuration property values

- ❑ if this commands are interpreted from a UC3 application 1.12.0 and higher this property pairs are merged to:
 - **configuration profiles** (related to application modes) or **independent configuration profiles** (independent from application modes)
 - **mode profiles** to allocate the created configuration profiles to this one which is used from the device number which was the target device in the old command
 - **device profiles** to allocate the device with the **mode profiles** and **independent configuration profiles** which have been merged based on the old configuration commands for this device number

1.2.1 Customer.ini parameter to accept old XML configuration commands

- ❑ the old XML commands can write values for configuration properties meant to be used only from one target device
- ❑ the new profile based approach for managing configuration data requests that all configuration data are distributed uniformly to all devices of the UC3 branch and finally allocated to a device in a device profile
- ❑ so there's a logical conflict between the old and the new handling for configuration data, because configuration properties used in old commands are grouped now in configuration profiles and one configuration profile can be used/shared in more mode profiles and device profiles
- ❑ therefore sending an old XML command which normally has only one device as target can be used now from a group of devices (may be all) because of sharing the modified configuration profile which is used from a group of devices via mode profiles and/or device profiles
- ❑ in order to avoid such logical behavior conflicts there's starting with application version 1.12.0 a new parameter in the customer.ini to define the configuration handling behavior
 - the new customer.ini parameter is in the section [network] (which will not be included in customer.ini profiles)
 - the new configuration parameter is:

[network]

bEnableConfigurationDataSynchronization = 0 or 1

- **bEnableConfigurationDataSynchronization = 1 (TRUE / default)**
 - ⇒ configuration data will be synchronized through basic data synchronization mechanism
 - ⇒ only the new TransUC3 commands can be used to transmit configuration data
 - ⇒ in this context when an old XML command for configuration is received there will be sent back a response containing an exception indicating that the command is not supported

- **bEnableConfigurationDataSynchronization = 0 (FALSE)**
 - ⇒ configuration data will be excluded from basic data synchronization mechanism and therefore handled only locally from each UC3 device
 - ⇒ in such context old XML commands for configuration data are accepted
 - ⇒ independent from the setting of this customer.ini parameter if a **configuration profile** or **independent configuration profile** is edited in NSM and then saved with the * button this information will be distributed to all devices in the UC3 branch, and so this setting will overwrite/synchronize existing configurations even if they were sent only locally to one device

1.2.2 Handling for old configuration XML commands with 'write' mode

- when one of the old configuration commands is received with the mode 'write' which is with application version 1.12.0 or higher a **configuration profile** (mode dependent) then is used following approach:
 - based on the configuration property name is identified where this property is contained in the profile based configuration structure to merge this settings to the database table for **configuration profiles** for the correct **application_mode** and **configuration_profile_type**
 - for the device number to which the old command is assigned to is searched in the database tables of **device profiles** and **mode profiles** if a **configuration profile** is still allocated to this device
 - ⇒ if that **configuration profile** exists and is found in the database table, the received value for the property name is used to update the existing value in the database
 - ⇒ if that **configuration profile** doesn't exist and is not found in the database table, then is created automatically an new **configuration profile** where the received property values are defined and this profile is assigned to the used **mode profile** (if necessary also new created) and **device profile** for the device used in the old command
- when one of the old configuration commands is received with the mode 'write' which is with application version 1.12.0 or higher a **independent configuration profile** then is used following approach:
 - based on the configuration property name is identified where this property is contained in the profile based configuration structure to merge this settings to the database table for **independent configuration profiles** for the correct **configuration_profile_type**
 - for the device number to which the old command is assigned to is searched in the database tables of **device profiles** if a **independent configuration profile** is still allocated to this device
 - ⇒ if that **independent configuration profile** exists and is found in the database table, the received value for the property name is used to update the existing value in the database

- ⇒ if that **independent configuration profile** doesn't exist and is not found in the database table, then is created automatically an new **independent configuration profile** where the received property values are defined and this profile is assigned to the **device profile** for the device used in the old command
- if no device profile record exists for the device to which the old command is sent to
 - ⇒ then first is created a **device profile** for the current device number
 - ⇒ for mode dependent configurations is created a **mode profile** as described above which is linked to the created **device profile**
 - ⇒ finally is created a
 - **configuration profile** for mode dependent configurations which is linked to the created **mode profile** (which is linked to the device profile)
 - **independent configuration profile** for mode independent configurations, which is linked to the created **device profile**

1.2.3 Handling for old configuration XML commands with 'read' mode

- when one of the old configuration command is received with mode 'read' for a configuration property name then is used following approach:
 - the configuration property name is identified to know if the corresponding property is part of a mode dependent **configuration profile** or of a **independent configuration profile**
 - ⇒ for mode dependent configurations the **application_mode** and the **configuration_profile_type** is used to know the **configuration profile** where the corresponding property is contained and which is allocated to the device number for which the request was sent from the old command
 - ⇒ for mode independent configurations the **configuration_profile_type** is used to know the **independent configuration profile** where the corresponding property is contained and which is allocated to the device number for which the request was sent from the old command
 - if that **configuration profile** or **independent configuration profile** exists and is found in the database then the value for the property name equivalent to received configuration property name is read out of the database and returned to the response file of the old XML command
 - if that **configuration profile** or **independent configuration profile** doesn't exists in the database there will be an exception in the response file.

1.2.4 Handling for old configuration XML commands with 'readall' mode

- when one of the old configuration commands is received with mode 'readall' then are returned values of properties based on the allocated **device-profiles**, **mode profiles** and **configuration profiles** or **independent configuration profiles** which are allocated to the current device and corresponding to configuration properties names supported by the old XML command