Breaking Changes

Breaking Changes in Webdesk 4.64

These notes are intended for Customer Admins and are relevant if IF6020, multiple employments, or customer-specific scripts, validators, reports, and Velocity templates are used.

1. Java 17 as a System Requirement

Webdesk 4.64 and the associated IF6020 TA connector run on Java 17. Java 11 is not sufficient for this version. Before the update, make sure that a supported Java 17 runtime is installed on the Webdesk server and on the TA connector and is enabled for the respective service.

If multiple Java installations are present, check in particular which Java version the Windows service or the startup script actually uses. Customer-specific Java jobs and integrations must also be tested after the update.

For an overview of the other new developments, see New Features in Webdesk 4.64[1].

2. Check Bean Names in Customer-Specific Scripts

The TA services were renamed per system for several TA systems. As a result, existing scripts, validations, reports, and Velocity templates may fail.

For IF6020, the following names in particular must be checked and adjusted:

- TaJournalService # Ta6020JournalService
- TaPersonService # Ta6020PersonService
- TaAccountService # Ta6020AccountService
- TaBookingService # Ta6020BookingService
- TaCorrectionService # Ta6020CorrectionService
- TaMetadataService # Ta6020MetadataService
- TaOptionsService # Ta6020OptionsService
- TaDefinitionService # Ta6020DefinitionService

Please search for all customer-specific usages of the old names, switch them to the appropriate system-specific bean names, and then test. For cross-system TA services, use the new generic bean name.

3. Change to stpersnr in IF6020

As of 4.64, stpersnr no longer automatically contains the Webdesk personnel number. When a TA person record is created, the TA ID is used there. If it is already taken, Webdesk generates a different unique ID.

The actual Webdesk employeeId is written to a separate, configured info field. Therefore, do not use stpersnr as the Webdesk personnel number anymore.

4. Configure Three Info Fields in IF6020

For the new mapping, three info fields must be set up and entered in tacon.properties:

- TaPersonDAO.nameOfFieldStoringPersonUid – Webdesk PersonUID
- TaPersonDAO.employeeIdField – actual Webdesk employeeId
- TaPersonDAO.effectiveEmploymentMarkerField – flag for the currently effective employment

The field names depend on the installation. As an example, the 4.64 documentation uses stpinfo151 for the PersonUID, stpinfo160 for the employeeId, and stpinfo30 for the effectiveness flag. Use the fields you have created. The info fields require at least the following lengths: PersonUID 50 characters, employeeId 20 characters (the Webdesk employeeId is at most 20 characters long), effectiveness flag 1 character.

Webdesk uses the flag J for a currently effective employment and N for an employment that is not effective.

Finding Free Info Fields (IF6020 on Microsoft SQL Server)

The following script lists all stpinfo fields of the IF6020 person master table (IF_PSTAMM) that contain no value in any record and are long enough for one of the three info fields. A field is considered empty if it is NULL in every record or contains only spaces or invisible characters. The script does not modify any data. It is run in the IF6020 database; the schema is determined automatically.

```
/* Empty stpinfo fields in the IF6020 person master (IF_PSTAMM) that can take the
   4.64 info fields:
     - TaPersonDAO.nameOfFieldStoringPersonUid    PersonUID    50+ characters
     - TaPersonDAO.employeeIdField                employeeId   20+ characters
 (PoPerson.employeeId is 20)
     - TaPersonDAO.effectiveEmploymentMarkerField flag J/N     1+ character
   Run in the IF6020 database (e.g. if6020). Read-only.
   Empty = every row is NULL or holds only spaces, tabs, line breaks,
   non-breaking spaces or NUL characters. */
SET NOCOUNT ON;

IF OBJECT_ID('tempdb..#stpinfo') IS NOT NULL DROP TABLE #stpinfo;
CREATE TABLE #stpinfo (
    table_name      nvarchar(300),
    column_name     sysname,
    data_type       sysname,
    max_length      int,
    total_rows      int,
    null_rows       int,
    blank_only_rows int,
    filled_rows     int);

DECLARE @schema sysname, @table sysname, @col sysname, @type sysname, @len int,
        @clean nvarchar(max), @sql nvarchar(max);

-- only text columns qualify; max_length -1 = (n)varchar(max)
DECLARE cols CURSOR LOCAL FAST_FORWARD FOR
    SELECT TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH
    FROM INFORMATION_SCHEMA.COLUMNS
    WHERE LOWER(TABLE_NAME) IN ('pstamm', 'if_pstamm')
      AND LOWER(COLUMN_NAME) LIKE 'stpinfo%'
      AND LOWER(DATA_TYPE) IN ('char', 'varchar', 'nchar', 'nvarchar', 'text', 'ntext')
      AND (CHARACTER_MAXIMUM_LENGTH = -1 OR CHARACTER_MAXIMUM_LENGTH >= 1);

OPEN cols;
FETCH NEXT FROM cols INTO @schema, @table, @col, @type, @len;
WHILE @@FETCH_STATUS = 0
BEGIN
    -- column value with spaces, tab, LF, CR, non-breaking space and NUL removed
    -- (binary collation, otherwise NUL cannot be replaced)
    SET @clean = N'REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE('
               + N'CAST(' + QUOTENAME(@col) + N' AS nvarchar(4000)) COLLATE
 Latin1_General_BIN'
               + N', N'' '', N''''), NCHAR(9), N''''), NCHAR(10), N''''), NCHAR(13),
 N''''), NCHAR(160), N''''), NCHAR(0), N'''')';

    SET @sql = N'INSERT INTO #stpinfo SELECT @schema + N''.'' + @table, @col, @type, @len,
 COUNT(*), '
             + N'SUM(CASE WHEN ' + QUOTENAME(@col) + N' IS NULL THEN 1 ELSE 0 END), '
             + N'SUM(CASE WHEN ' + QUOTENAME(@col) + N' IS NOT NULL AND LEN(' + @clean +
 N') = 0 THEN 1 ELSE 0 END), '
             + N'SUM(CASE WHEN LEN(' + @clean + N') > 0 THEN 1 ELSE 0 END) '
             + N'FROM ' + QUOTENAME(@schema) + N'.' + QUOTENAME(@table);

    EXEC sp_executesql @sql,
         N'@schema sysname, @table sysname, @col sysname, @type sysname, @len int',
         @schema, @table, @col, @type, @len;

    FETCH NEXT FROM cols INTO @schema, @table, @col, @type, @len;
END
CLOSE cols;
DEALLOCATE cols;

IF NOT EXISTS (SELECT 1 FROM #stpinfo)
BEGIN
```

```
    SELECT DB_NAME() AS current_database,
           'No IF_PSTAMM / pstamm table with text stpinfo columns visible in this database'
 AS problem;
END
ELSE
BEGIN
    SELECT CASE WHEN max_length = -1 OR max_length >= 50 THEN 'PersonUID, employeeId or
 flag'
                WHEN max_length >= 20                    THEN 'employeeId or flag'
                ELSE                                          'flag only'
           END AS usable_for,
           column_name, data_type, max_length, total_rows, null_rows, blank_only_rows
    FROM #stpinfo
    WHERE filled_rows = 0
    ORDER BY CASE WHEN max_length = -1 OR max_length >= 50 THEN 0
                  WHEN max_length >= 20                    THEN 1
                  ELSE                                          2 END,
             max_length, LEN(column_name), column_name;
END
```

The usable_for column shows what a field is long enough for: PersonUID, employeeId or flag (50 characters or more), employeeId or flag (20 to 49 characters), or flag only (fewer than 20 characters). Within each group, the shortest fields are listed first.

Empty only means that IF_PSTAMM contains no value. Before using the field, verify that it is not used in customer-specific payroll scripts, PLIs, or forms of the 6020 client.

5. Migrating Existing Data

1. Configure the three info fields and the associated properties.
2. Migrate existing TA person records with the supplied TA update V48 or the job migrateEmployeeId. This writes the stable Webdesk employeeId and the effectiveness flag to the configured info fields.
3. Use the option to overwrite existing info field values only after prior verification.
4. Write the existing PersonUIDs to the configured 6020 info fields using the Webdesk admin action save_personuids_into_TaPersons. For newly created TA person records, the PersonUID is written automatically once configuration is correct.

6. TA ID in the HRX Employments

With multiple employments, each HrPersonEmployment requires the matching TA ID back-reference in taId. A Webdesk person remains the central master record; in TA, a separate TA person record is maintained for each employment.

Webdesk 4.64 includes upgrade V122 for this. It synchronizes the person's TA ID with the matching currently valid employment. Because the upgrade does not abort completely on an error, the results and warnings must be reviewed. With multiple employments, each TA ID must be uniquely assigned to the correct employment. Do not change missing or incorrect assignments directly in the database; correct them using the provided Webdesk functions or with Workflow support.

7. Verification After the Upgrade

- Check the Java version of the Webdesk service and of the TA connector.
- Test parallel employments, termination, and re-entry.
- Verify that the currently effective employment is flagged with J and employments that are not effective are flagged with N.
- Verify that the Webdesk employeeId remains stable in the separate info field and that stpersnr is no longer used for it.
- Test the TA journal, correction and absence client, as well as customer-specific scripts, reports, and validations.

If there is no IF6020 integration, no multiple employments, and no customer-specific TA references, there is usually no need for manual adjustments for these items.

1. /daisy/personalwolke-admin/11418-dsy.html?language=4