6.04.06 SynR3.js Synchronization Function

The synchronization function itself is, as was previously explained, described in SynR3.js. These are the generic functions used to execute upload and download functionality. See the Data Synchronization section for more complete details. There is also a white paper defining the ECMA script language that addresses the generic functions. This section contains the code for the generic synchornization functions, as shown below.

The four functions are:

- SR3GenericUploadProcess
- SR3GenericUploadProcessConfigurable
- SR3GenericDownloadProcess
- SR3GenericDownloadProcessConfigurable

SR3GenericUploadProcess

To use the generic upload function that uploads data to the SAP application server, it is necessary to properly define the functions in the script. The generic upload function takes the following parameters:

- **strUploadCode:** A required parameter that specifies the specific transaction code to be uploaded to the target SAP application server.
- **iShowRows:** Specifies how many rows will be uploaded to the SAP application server.
- **strAdditionalCondition:** Specifies if there are any added conditions to the upload.
- **strAdditionalTableNames:** If the upload requires additikonal tables to input data into, this optional parameter enables users to specify those table names in the upload function.
- **bSendStatusMessage:** An optional Boolean parameter that will send status messages to the user during the upload process.

The syntax for the generic upload process is as follows:

```
Array SR3GenericUploadProcess(strUploadCode,iShowRows,strAdditionalCondition,strAdditionalTableNames,bSendStatusMessage);
```

An example of the GenericUploadProcess in use is shown below. In this example, neither the additional table name nor the send status message parameter is used.

```
arResult = SR3GenericUploadProcess.call(this,"mm01",10,"condition");
```

SR3GenericUploadProcessConfigurable

To use the generic configurable upload function that uploads data to the SAP application server, it is necessary to properly define the functions in the script. The generic upload function takes the following optional parameters:

- **objServer:** Specifies to which SAP server the data will be uploaded.
- **strUserName:** The username for the current user.
- **strPassword:** The SAP password for the current user.
- **strUploadCode:** Specifies the specific transaction code to be uploaded to the target SAP application server.
- **iShowRows:** Specifies how many rows will be uploaded to the SAP application server.
- **strAdditionalCondition:** Specifies if there are any added conditions to the upload.
- **strAdditionalTableNames:** If the upload requires additional tables to input data into, this parameter enables users to specify those table names in the upload function.
- **bSendStatusMessage:** A Boolean that will send status messages to the user during the upload process.

The syntax for the generic upload process is as follows:

```
Array SR3GenericUPloadProcessConfigurable(objServer,strUserName,strPassword,strUploadCode,iShowRows,strAdditionalCondition,strAdditionalTableNames,bSendStatusMessage);
```

An example of the GenericUploadProcess in use is shown below. In this example, neither the additional table name nor the send status message parameter is used.

```
objServer = {srevername:"serverName",client:"800",language:"EN"};
arResult = SR3GenericUploadProcessConfigurable.call(this,objServer,"name","password","mm01",10,"condition");
```

SR3GenericDownloadProcess

To use the generic upload function that uploads data to the SAP application server, it is necessary to properly define the functions in the script. The generic upload function takes the following parameters:

- **strDownloadCode:** A required parameter that specifies the specific transaction code to be uploaded to the target SAP application server.
- **arCriteriaScreen**: Specifies how many rows will be uploaded to the SAP application server.
- **arKeyFields:** Specifies the key fields to be downloaded from the SAP apoplciation server.
- **strAdditionalCondition:** Specifies if there are any added conditions to the

upload.
- **bSendStatusMessage:** An optional Boolean parameter that will send status messages to the user during the upload process.
- **strDownloadTable:** Optional parameter specifying the specific tables in a given transaction to be downloaded.

The syntax for the generic download process is as follows:

```
Array SR3GenericDownloadProcess(strDownloadCode,arCriteriaScreen,arKey
Fields,bSendStatusMessage,strDownloadTable);
```

An example of the GenericDownloadProcess in use is shown below. In this example, neither the send status message nor the DownloadTable parameters are used.

```
arResult = SR3GenericDownloadProcess.call(this,"wkdr",arCriteriaScreen
,["g_wo_no", "g_wo_activity"]);
arResult = SR3GenericDownloadProcess.call(this,"wkdr",arCriteriaScreen
,["g_wo_no", "g_wo_activity"]);
```

SR3GenericDownloadProcessConfigurable

To use the generic configurable upload function that uploads data to the SAP application server, it is necessary to properly define the functions in the script. The generic upload function takes the following optional parameters:

- **objServer:** Specifies to which SAP server the data will be uploaded.
- **strUserName:** The username for the current user.
- **strPassword:** The SAP password for the current user.
- **strDownloadCode:** Specifies the specific transaction code to be uploaded to the target SAP application server.
- **arCriteriaScreen:** Specifies how many rows will be uploaded to the SAP application server.
- **arKeyFields:** Specifies the key fields to be downloaded from the SAP apoplciation server.
- **strAdditionalCondition:** Specifies if there are any added conditions to the upload.
- **bSendStatusMessage:** An optional Boolean parameter that will send status messages to the user during the upload process.
- **strDownloadTable:** Optional parameter specifying the specific tables in a given transaction to be downloaded.
- **bType:** Optional parameter that specifies which records are to be downloaded. This parameter has four possible conidtions, as follows:
    - 0: Default setting. Modified records (='X') are skipped and only existing records are updated.
    - 1: All records are deleted before the download begins.

  - 2: Modified records are deleted before the download begins and existing records are updated (delete + add).
  - 3: Unmodified records (<>'X') are deleted before the download begins and existing records are skipped.

The syntax for the configurable generic download process is as follows:

```
Array SR3GenericDownloadProcessConfigurable(objServer,strUserName,strPassword,strDownloadCode,arCriteriaScreen,arKeyFields,bSendStatusMessage,strDownloadTable,nType);
```

An example of the GenericUploadProcess in use is shown below. In this example, the SendStatusMessage parameter is turned off, there is no download table specified and the type is set to '1' as described above.

```
objServer = {srevername:"serverName",client:"800",language:"EN"};
arResult = SR3GenericDownloadProcessConfigurable.call(this,objServer,"name","password","wkdr",arCriteriaScreen,["g_wo_no","g_wo_activity"],false," ",1);
```

Unique solution ID: #2100
Author: sarvani.kusuri@guixt.com
Last update: 2021-06-03 18:44