

Utility Classes - Overview

Version 14 service pack 9

3	Revision History	4
3.1	Changes since SPS8	4
5	Transport contents	6
5.1	Overview	6
5.2	Example and demo programs	6
5.3	Classes	7
6	Class overview	9
6.1	ZCL_DT_APPSERVER	9
6.2	ZCL_DT_BACKGROUNDJOB	9
6.3	ZCL_DT_BGJOB_ERRORHANDLING	9
6.4	ZCL_DT_CONSTANTS	11
6.5	ZCL_DT_CONVERSIONS	11
6.6	ZCL_DT_EMAILRECIPIENTS	12
6.7	ZCL_DT_EXTERNALCOMMANDS	13
6.8	ZCL_DT_FILE	13
6.9	ZCL_DT_FTP_SAP	15
6.1	ZCL_DT_FTP_SCRIPT	16
6.2	ZCL_DT_HTML	16
6.3	ZCL_DT_LISTOUTPUT	16
6.4	ZCL_DT_LOGICALFILENAMES	17
6.5	ZCL_DT_LOGICALFILEPATHS	18
6.6	ZCL_DT_MAIL	18
6.7	ZCL_DT_MESSAGE	19
6.8	ZCL_DT_RUNTIMEMESSAGE	20
6.9	ZCL_DT_SPREADSHEET	20
6.10	ZCL_DT_TRANSPORTS	21
6.11	ZCL_DT_UTILITIES	22
6.12	ZCL_DT_XML	22
7	Table Maintenance	23

1 Rational

The classes (of which there are twenty one) have many uses; they can be used for file handling FTP, field conversions etc, and all use the new exception-handling concept implemented in SAP basis 6.20. Let's take the file handling class as an example. How many times have you written a program which reads/writes a file to the application server? This invariably involves you having to use the ABAP dataset statement and pass each line to the file at a time, handling exceptions along the way. And what happens the next time you need this function, you most likely re-code a new version changing the error handling and messaging as you go.

With the file class you can upload a file from the application server with two method calls, a total of six lines of code. This incorporates both exception and success message handling. And on top of this provides one point of contact if the code needs to be modified to accommodate any sap enhancements such as Unicode.

The classes are non-dependant on the operating system they are installed on. The coding uses external commands, logical file paths & names to keep application server interaction consistent.

The solution also incorporates example programs that can be used as a basis for training purposes and most importantly as a driver programs to aid the testing of changes to the classes.

2 Revision History

2.1 Changes since SPS8

- Class ZCL_DT_BACKGROUND. The class was incorrectly specified as “public”, it has now been changed to “Abstract”.
- New method – PICKTRANSPORT added to class ZCL_DT_TRANSPORTS.
- Fix – In method ZCL_DT_LOGICALFILENAME=>PICKFILENAMES, which prevented any filenames being returned
- New Method – ZCL_DT_CONVERSIONS=>FORMATCASEORPAD. If the passed in value contains ALPHANUMERIC then convert to upper case. If the field contains NUMERIC only then pad with leading zeros.
- Class ZCL_DT_FTP split into two new classes, ZCL_DT_FTP_SAP and ZCL_DT_FTP_SCRIPT to make them more understandable and useable.
- Updated Method – ZCL_DT_SPREADSHEET=>CREATECONTENTS. Allows different formatting of decimal contents to the one specified in the user master settings. E.g a spreadhseet may have been created in german but loaded in SAP by a user with English settings.
- Added Logical filepath ZDT_TEMPORARY_PATH to the transport as it was missing and causing new instances of class ZCL_DT_FILE to fail with an error message.
- Updated Method - ZCL_DT_UTILITIES=ISDEBUGGINGENABLED. Changed the memory ID from X to the current running program.

3 Naming conventions

The following SAP object naming conventions are used throughout the implementation. The naming convention is easy to read and in most cases will allow the objects to be imported into a client system without naming conflicts occurring. Here is a list of all the naming standards used throughout the solution:

Convention	Used for
SAPMZDT_*	Module Pool
ZDTP_*	Programs
ZDTINC_*	Includes
ZCL_DT_*	Global classes
ZCX_DT*	Global exception classes
ZDTM_*	Message classes
Z_DT_*	Function modules
ZDTFG_*	Function groups
ZDT_*	DD tables
ZDTT_*	DD: Table types
ZDTS_*	DD: Structures
ZDTD_*	DD: Data elements
ZDTDOM_*	DD: Domains
ZDTV_*	DD: Views
ZDTXSLT_*	XSLT programs
ZDT_*	Transaction codes
ZDT_*	Packages
ZDT_*	Logical file paths & names
ZDTX_*	External commands

All of the objects are held within the package "ZDT_UTILITYCLASSES", which is a sub package of ZDT_DEVELOPMENT

4 Transport contents

4.1 Overview

There are two transports for the utility classes, the first contains the classes themselves, and the second contains a sub-package with all the demo programs listed.

Transports	
Transport type	Transport Name
Utility Classes	APDK90035
Demo Programs	APDK90043

The demo programs transport is dependent on the utility classes transport being installed.

4.2 Example and demo programs

There are 15 programs included with the utility classes, 14 of these are used for testing/demonstrating and one is a report. Below is a list of all the programs and their roles:

Demo Programs	
Program name	Description
ZDTP_COPYFILEFROMSAP	Copy a file from the SAP server to the user local PC
ZDTP_COPYFILETOSAP	Copy a file from the users Pc to the SAP server
ZDTP_EMAIL_DEMO	Test the e-mail class
ZDTP_FILE_DEMO	Tests the file interface class
ZDTP_FTPFILEFROMSAP	FTP a file from SAP through a script
ZDTP_FTPFILETOSAP	FTP a file to SAP through a script
ZDTP_JOB_DEMO	Background job demo program
ZDTP_LOGICALFILE_DEMO	Tests the logical file class
ZDTP_SPREADSHEET_DEMO	Tests the spreadsheet class
ZDTP_SAPFTP_COMMAND_DEMO	Tests the FTP class
ZDTP_SAPFTP_COPY_DEMO	Tests the FTP class

ZDTP_SAPFTP_HELP_DEMO	Tests the FTP class
ZDTP_SAPFTP_R3COPY_DEMO	Tests the FTP class
ZDTP_SAPFTP_VERSION_DEMO	Tests the FTP class

Reports	
Program name	Description
ZDTP_BACKGROUNDJOBLOG	Report to show the messages stored in the DB from the background job logging class

4.3 Classes

There are eighteen utility classes in total, which are summarised below.

Utility classes	
Class name	Description
ZCL_DT_APPSERVER	Directory listing & picking
ZCL_DT_BACKGROUNDJOB	Creating and deleting of background jobs in the scheduler
ZCL_DT_BGJOB_ERRORHANDLING	Messaging of both foreground and background jobs and controlled cancelling of background jobs.
ZCL_DT_CONSTANTS	Constants used by all the classes
ZCL_DT_CONVERSIONS	Data conversion
ZCL_DT_EMAILRECIPEINTS	E-mail recipients for the mail class
ZCL_DT_EXTERNALCOMMANDS	External command handling
ZCL_DT_FILE	File handling on the application server and local PC
ZCL_DT_FTP	FTP of files between SAP systems
ZCL_DT_HTML	Conversion of structures/tables to HTML
ZCL_DT_LISTOUTPUT	Adds user defined selection screen details to an ABAP list.
ZCL_DT_LOGICALFILENAMEAMES	Handles all aspects of logical file names.
ZCL_DT_LOGICALFILEPATHS	Handles all aspects of logical file paths.
ZCL_DT_MAIL	E-mail sending from ABAP both text and binary.
ZCL_DT_MESSAGE	Visual messages to the user

ZCL_DT_RUNTIMEMESSAGE	Generated message in an object
ZCL_DT_SPREADSHEET	Spreadsheet integration
ZCL_DT_TRANSPORTS	Get information about the transport system
ZCL_DT_UTILITIES	Collection of useful methods
ZCL_DT_XML	Convert data to XML or HTML via XSLT

5 Class overview

5.1 ZCL_DT_APPSERVER

The APPSERVER class is used for activities on the SAP application server. Currently it has the ability to check if a directory exists, and secondly to allow the contents of a directory to be listed and returned to an ABAP program, or presented to a user as a pop up window from which the user can choose one or more files.

<i>Static class</i>	
<i>Method</i>	<i>Description</i>
AddTrailingSlash	Adds a trailing slash to a file path
CheckDirExists	Checks to see if a specified directory exists.
ListFiles	Returns a list of files in a directory to the ABAP program.
PickFiles	Allows a user to pick one or more files form a directory.

5.2 ZCL_DT_BACKGROUNDJOB

This class is used to manage jobs in the SAP standard job scheduler SM36.

<i>Static class</i>	
<i>Method</i>	<i>Description</i>
CancelJob	Cancel a running background job
CreateBackgroundJob	Create a new background job
DeleteBackgroundJob	Delete a background job
DoesJobExist	Does a job exist with the specified name
GetJobInfo	Get the job info of the current running background job

5.3 ZCL_DT_BGJOB_ERRORHANDLING

The Job class handles any messaging and errors generated by a program whether within a foreground run or a background job. Each program instantiates a job object, against this object we can add messages.

In addition to receiving and storing messages in the SAP database the object can also send either success or error e-mails. Further if the program is running in background the object has the ability to terminate the current background job. This might be useful if further job steps mustn't run without the previous one completing successfully.

The object can take two courses of action, either

- **The program fails.** Any messages stored against the job object can be saved to the database, the object can choose to send either a problem or cancel e-mail to a user or distribution group. If running in background the background job can then be cancelled.
- **The program works.** Any messages stored against the job object can be saved to the database; the object can choose to send a success e-mail to a user or distribution group. The program finishes and the next program in the job instantiates a new object.

<i>Instance class</i>	
<i>Method</i>	<i>Description</i>
AddExceptionObjMessage	Adds an exception object message to the background job
AddMessage	Add a string based message to the background job
AddRuntimeMessage	Adds a runtime message to the background job
AddT100Message	Add a T100 based message to the background job
Cancel	Cancel an active background job
Constructor	Instantiate this class
DeleteAllMessages	Delete all messages generated for a background job
GetNumberOfMessages	Get the current number of messages for the background job
IsAlertPercentage	Has the alert percentage been met? This is useful if the job is supposed to cancel after X percentage failures.
Save	Save all messages to the DB message logging tables
SendEmail	Send success, problem or cancellation e-mail.
SortAndCompress	Sort and compress the object message to get rid of duplicates

Any messages stored against the job objects can be viewed with transaction ZDT_JOBLOG. The messages are stored for a period of 60 days before deletion.

5.4 ZCL_DT_CONSTANTS

This class holds all of the global constants used within the utility classes. It has no methods. Example constants are TRUE & FALSE.

<i>Abstract class</i>		
<i>Constant</i>	<i>Description</i>	<i>Value</i>
TRUE		X
FALSE		
STATUSCHANGED		CHANGED
STATUSNEW		NEW
STATUSERROR		ERROR
STATUSPOSTED		POSTED
STATUSSAVED		SAVED
STATUSDELETED		DELETED
BAPIERROR		E
BAPISUCCESS		S
BAPIINFO		I
BAPIWARNING		W

5.5 ZCL_DT_CONVERSIONS

This class performs conversions on data, either converting single fields, adding values into string fields or converting internal tables to/from CSV to/from internal ABAP format.

<i>Static class</i>	
<i>Method</i>	<i>Description</i>
AppendControlCharacter	Append a hexadecimal value to the end of a string field.
InsertControlCharacter	Insert a hexadecimal value into a string at a certain offset.
PadWithLeadingZeros	Pad a field out with leading zeros.
RemoveLeadingZeros	Remove leading zeros from a field.
SplitStringAtFixedLength	Split a string field at a fixed length into a table of type string.

Static class	
ConvertDateToExternal	Convert a date from internal to external format.
ConvertDatetoInternal	Convert a date from external to internal format.
ConvertCSVToItab	Convert a string table full of CSV data to an internal table based on an internal ABAP structure.
ConvertItabToCSV	Convert an internal table to a table of CSV data. Can also add a header line o the CSV with the name of the ABAP fields. Can call multiple times to append more than one ITAB to the CSV data.
ConvertCSVToWa	Convert a string of CSV data to an ABAP work area.
ConvertWaToCSV	Convert an ABAP work area to CSV format.
XstringToSolix_Tab	Converts a binary string into a table of type solix_tab.
FormatCaseOrPad	Converts any alphas to uppercase. Pads withleading zeros if contains numbers.

5.6 ZCL_DT_EMAILRECIPIENTS

This class is used for handling email recipients for the class ZCL_DT_MAIL. It is used to guarantee data integrity for passing recipients into the mail class. Validity checks are carried out against the e-mail addresses added.

Instance class	
Method	Description
AddBCCRecipients	Add a BCC recipient to the object
AddCCRecipient	Add a CC recipient to the object
AddRecipient	Add a recipient to the object
AddSapDistributionList	Add all the members of an SAP distribution list
CheckDistListExists	Check that the SAP distribution list exists
CheckEmailAddressValidity	Check to see if the e-mail address is constructed properly
ClearSAPDistLists	Clear the object of all SAP distribution lists
ClearMailRecipients	Clear all mail recipients out of the object
getBCCRecipients	Get BCC recipients from the object
getCCRecipient	Get CC recipient from the object
getRecipient	Get recipients from the object

<i>Instance class</i>	
pickDistributionList	Allow a user to choose one of the distribution lists defined in SO23. This can be restricted to only private lists if required.

The SAP distribution list can be created within transaction SO23, users can be defined as recipients, CC recipients and BCC recipients.

5.7 ZCL_DT_EXTERNALCOMMANDS

This class encapsulates SAPs' standard approach for calling external commands on the SAP server and provide a standard error framework for handling exceptions.

<i>Static class</i>		
<i>Method</i>	<i>Description</i>	
CallExternalCommand	Calls an external operating system command on the SAP server	
Constant	Description	Value
COMMANDLIST	List all file sin a directory	ZDTX_LIST
COMMANDCOMPRESS	Compress files into an archive	ZDTX_COMPRESS
COMMANDCOPY	Copy a file	ZDTX_COPY
COMMANDDECOMPRESS	De-compress an archive file	ZDTX_DECOMPRESS
COMMANDMOVE	Move a file	ZDTX_MOVE
COMMANDRENAME	Rename a file	ZDTX_RENAME
COMMANDFTP	FTP a file to another server	ZDTX_FTP_PUT
COMMANDSFTP	FTP a file from another server	ZDTX_FTP_GET
COMMANDMKDIR	Make a directory	ZDTX_MKDIR

5.8 ZCL_DT_FILE

The file class was designed to make file handling simpler. The methods in this class apply to both files on the SAP server and the local PC. Of all the classes detailed in this document this class is the most useful and generally the most frequently used. It was created to simplify file handling and most importantly error handling. It works by abstracting both the dataset and GUI commands and provides a simple interface for file interaction.

Since all the methods apply both to the application server and PC it is possible to change the file processing location by a simple flag on the constructing factory method. No other coding is necessary.

The class derives if the file handling is taking place on either an UNIX or Windows platform (both on the application server and local PC) and handles operating system calls accordingly.

The other advantage of the class is that it reduces the impact point of using SAPs GUI methods, and any changes or updates only need to be completed in one place and not in every calling program.

The class can also use the logical file path 'ZDT_PATH_RESTRICTION' to limit access to a specific directory and its subdirectories on the application server.

<i>Instance class</i>	
<i>Method</i>	<i>Description</i>
AddTrailingSlash	Add a trailing slash to a file path
ArchiveFile	Archives the file into the specified directory, with or without a date/timestamp. The source file can also be deleted
ArchiveFileUsingLogical	Archive a file using a destination logical file path
CheckfileExists	Check the file exists in the location specified.
CompressFile	Compress the file (only applies to the application server)
Copyfile	From one location to another.
CopyToPC	Copy a file on the SAP sever to the local computer
CopyToSAP	Copy a file on the local computer to the SAP server.
CreateByLogicalFilename	Create a file instance based upon a logical filename
CreateByLogicalFilepath	Create a file instance based upon a logical file path and a physical filename
CreateByPhysicalFilename	Create a file instance based upon a physical path and physical filename
CreateByCompletePathAdnFile	Create a file instance based upon a concatenated physical file path and physical filename.
DeleteFile	Delete the specified file
DeleteTemporaryFile	Delete a file in the temporary file directory
DetermineFilesystem	Determine the underlying server file system type
DetermineOPsystem	Determine the server operating system.
Download	Download the file, with or without confirmation
FindTempPath	Retrieves temp path from logical file path ZDT_TEMPORARY_PATH.

GetPathRestriction	Get the path restriction for file operations. Any file activity cannot take place above this path.
MoveFile	Form one directory to another
PadFilename	Pads a filename and cycle count out with zeros. Result = filename + zeros + cycle count.
RenameFile	Renames a file
SetAsciiFile	Set the file to be of type ASCII.
SetBinaryFile	Set the file type to be binary
SetEncoding	Set the file encoding type
SetLegacyCodePage	Set the legacy codepage of the file
SetTabDelimited	Set the ASCII file to be tab delimited (Only applies to GUI files).
SplitFilename	Split a filename or path + filename into all its constituent parts.
UncompressFile	Uncompress a zip file (only applies to the application server)
Upload	Upload a file, from line/all lines, in blocks of X.
UnzipFile	Unzip a specified file
ZipFile	Zip a specified file

5.9 ZCL_DT_FTP_SAP

This class encapsulates the standard SAP FTP commands. FTP connection details can be stored using transaction ZDT_FTPVAR & ZDT_VARDET ready to be used in any FTP method calls.

<i>Instance needed for SAP FTP</i>	
<i>Needed if using the standard SAP FTP functionality</i>	
<i>Method</i>	<i>Description</i>
Constructor	Instantiate the object
FTPCommand	Pass a command to the R/3 FTP server
FTPDisconnect	Disconnect from the R/3 FTP server
FTPCommandList	List of available FTP commands on the internal R/3 FTP server
FTPCopy	Copy a file from this R/3 server to another R/3 server.
FTPSToR3Blob	FTP a binary file from another R/3 server into an internal table
FTPSToR3Text	FTP a text file from another R/3 server into an internal table
ExecuteScript	Execute a script of FTP commands between the two R/3 systems

FTPR3ToServerText	FTP a text table from this R/3 server to another R/3 server as a file.
FTPR3ToServerBlob	FTP a binary table from this R/3 server to another R/3 server as a file.
SetFilename	Set the name of the file we are dealing with.
SetFilePath	Set the file path of the file we are dealing with.
FTPVersion	Get the version of the SAP R/3 FTP command.

5.1 ZCL_DT_FTP_SCRIPT

This class provides a method of calling external scripts on the SAP application server. FTP connection details can be stored using transaction ZDT_FTPVAR & ZDT_VARDET ready to be used in any FTP method calls.

Static methods for User scripted FTP	
Needed if using user-defined script based functionality	
Method	Description
FTPFILE	FTP a file to/from SAP based upon details in the variants table.

5.2 ZCL_DT_HTML

The HTML class deals with creating HTML; in the example below it is possible to convert an internal table to HTML. The HTML generated is W3C compatible, although does not generate in-line CSS style sheet data.

Static class	
Method	Description
ConvertItabToHTML	Convert an internal table to HTML. If the table is based upon a DD structure then you can add column headings, either using technical DD field names or by adding DD descriptive texts.

5.3 ZCL_DT_LISTOUTPUT

The LISTOUTPUT class is used within the print event receiver methods of ALV processing. When a user chooses to print from an ALV grid the following methods are called:

- print_end_of_list
- print_end_of_page
- print_top_of_list
- print_top_of_page

Within these methods it is possible to influence the printed version of the report and add custom text and variables in the headers and footers. It is also possible to add any report select-options and parameters and have them displayed with their values on the report header.

<i>Instance class</i>	
<i>Method</i>	<i>Description</i>
Constructor	Instantiates the object
PrintListFooter	Writes user defined text at the end of the printout.
PrintListHeader	Writes user and system details to the start of the report, plus additional program defined content.
PrintPageFooter	Print an additional text on the footer of every page
PrintPageHeader	Prints the report name and current page number on the header of each report page.
PrintParameter	Add a selection screen parameter and its values to the report printout.
PrintSelectOption	Add a selection screen select-option and its values to the report output.

5.4 ZCL_DT_LOGICALFILENAMES

Logical filenames have been implemented in SAP for years but are very rarely implemented on customer sites. This class has adds functionality which is important for end users to use pick the required logical filename they wish to use and for developers to convert the logical file names to operating system level filenames and corresponding file paths. This class can be used directly by the ZCL_DT_FILE class. An example of implementing this class can be seen within program ZDTP_LOGICALFILE_DEMO.

<i>Static class</i>	
<i>Method</i>	<i>Description</i>
FindAllFilenames	Find all logical filenames and return them in an internal table
RestrictFilenames	Restrict the list of logical filenames we have returned from method FindAllFilenames().
Pickfilename	Pick a logical filename from a F4 drop down.
GetPhysicalPath	From a logical filename derive the physical file path being used

<i>Static class</i>	
getPhysicalFilename	From a logical filename derive the physical file, its type, and the physical filepath it uses.
GetLogicalPath	From a logical filename derive the path it is using

5.5 ZCL_DT_LOGICALFILEPATHS

Logical file paths (like logical filenames) have been implemented in SAP for years but are very rarely implemented on customer sites. This class has adds functionality which is important for end users to use pick the required logical path they wish to use and for developers to convert the logical file paths to operating system level file paths. This class can be used directly by the ZCL_DT_FILE class. An example of implementing this class can be seen within program ZDTP_LOGICALFILE_DEMO.

<i>Static class</i>	
<i>Method</i>	<i>Description</i>
FindAllPaths	Find all logical file paths and return them in an internal table
RestrictPaths	Restrict the list of logical file paths we have returned from method FindAllPaths().
Pickfilepath	Pick a logical file path from an F4 drop down.
GetPhysicalPath	Determine a physical file path form a logical file path.

5.6 ZCL_DT_MAIL

Sending e-mails from within SAP has never been particularly easy. Especially when you wish to add attachments, or send HTML or binary messages. However the MAIL class has been designed with all of these scenarios in mind.

<i>Instance class</i>	
<i>Method</i>	<i>Description</i>
CreateBySenderAddress	Create an e-mail instance with reply to user defined
CreateByUsername	Create an e-mail instance with reply to derived from the current users username.
CreateMailBody	Create the mail body as TEXT, HTML or BINARY.
AddAttachment	Add a TEXT, HTML or BINARY attachment to the mail. You do not need to add a mail body to add an attachment.

<i>Instance class</i>	
SetTrace	Set the SCOT trace on for this e-mail. Ideal for problem solving.
SetExpiresOn	Set the date on which the e-mail expires.
SetPriority	Set the mail priority.
Send	Send the e-mail immediately or part of a batch job
SendHTMLMail	Static method: Send a simple HTML based e-mail without having to call all the methods above.
SendTextMail	Static method: Send a simple text based e-mail without having to call all the methods above.
SendHTMLMailFromDDusingXSLT	Static method: Combines an internal table with an XSLT program to generate an e-mail without having to call all the methods above.
SendHTMLMailFromXMLUsingXSLT	Static method: Combines an XML structure with an XSLT program to generate an e-mail without having to call all the methods above.

5.7 ZCL_DT_MESSAGE

Static class used for converting messages or statuses from one format to another.

<i>Static class</i>	
<i>Method</i>	<i>Description</i>
BapiStatusToALVIcon	Converts a BAPI status to ALV icon ICON_LED_GREEN, ICON_LED_YELLOW and ICON_LED_RED.
BapiStatusToALVstatus	Convert a table of BAPI result and updates their statuses for ALV display. BAPI status (I,E,W,S) to ALV status (1,2,3).
BapiStatusToALVstatusSingle	Convert a BAPI result and updates the status for ALV display. BAPI status (I,E,W,S) to ALV status (1,2,3).
BooleanStatusToALVIcon	Converts TRUE to ALV icon ICON_LED_GREEN, and FALSE to error ALV icon ICON_LED_RED.
BooleanStatusToALVStatus	Converts TRUE to ALV status success (3), and FALSE to error ALV status error (1).
MessageStatusToAlvStatus	Convert a single message (I,E,W,S) to ALV status (1,2,3).
MessageToBapiMessage	Convert a T100 based message with parameters to a type of BAPI_MSG
MessageToString	Convert a T100 based message with parameters to a string
TextToString	Convert a program text with parameters to a string. Useful if your base

<i>Static class</i>	
	message needs to be longer than a T100 message of 72 chars.
UpdateSAPGUIIndicator	Calls the function to display the SAPGUI progress indicator

5.8 ZCL_DT_RUNTIMEMESSAGE

This class is extremely useful it allows a message to be passed back from a method without displaying it on the screen. When coding methods it is important that we allow the calling program to decide the program flow. An example of this is if we wish to check the outcome of a logical test.

- Call method 1
- Call method 2
- Decide what to do – then display message

If we wished to do this and the first method produced a message of type 'W' then the program would raise a warning message within the program and this would be incorrect.

Raising a message statement within a globally shared method is incorrect, as we don't know how the calling object intends to handle it.

With the runtime message class we can return or export a message as an object and let the calling program handle it.

<i>Instance class</i>	
<i>Method</i>	<i>Description</i>
Constructor	Pass in the message type (I,E,W,S), T100 message class, message number and up to 4 message parameters.

5.9 ZCL_DT_SPREADSHEET

This class encapsulates the standard SAP OLE spreadsheet implementation to allow you to either start MS Excel independently or embedded into the SAPGUI. An existing spreadsheet can be uploaded on start or a new blank spreadsheet created. An example program to demonstrate the usage of this is available it is called ZDTP_SPREADSHEET_DEMO.

<i>Instance class</i>	
<i>Method</i>	<i>Description</i>
CreateUsingCustContainer	Create an instance of the spreadsheet object based on a SAPGUI custom container. Either start with a blank spreadsheet, or pop-up a file picker window, or pass in a path and file name and the spreadsheet will automatically uploaded.
CreateUsingGuiContainer	Create an instance of the spreadsheet object based on a SAPGUI gui container. Either start with a blank spreadsheet, or pop-up a file picker window, or pass in a path and file name and the spreadsheet will automatically uploaded.
AddComment	Add a comment to a single cell
AddComments	Add a comment to multiple cells
ClearCommentsFromRange	Clear all the comments from a pre-defined range
CloseDocument	Close the document and if necessary prompt to save.
ColourCell	Colour a spreadsheet cell
GetDataFromCurrentSheet	Get all the data from the current sheet
GetDataFromRange	Get all the data from a pre-defined range in the current sheet
HasWorkbookChanged	Has the workbook changed since loading
InsertTableIntoSheet	Insert an internal table into the current sheet
InsertTableIntoRange	Insert an internal table into a pre-defined range in the spreadsheet
SetCurrentSheet	Set a sheet to be displayed as the current sheet
SetFlushingIndicator	Tell SAP to send the previous method calls to the front-end
<i>Interface</i>	<i>Description</i>
i_oi_spreadsheet	Additional methods available for spreadsheet interaction. Accessible by objSpreadsheet->objWorkSheet->method name

5.10 ZCL_DT_TRANSPORTS

Information on the current transport system.

<i>Static class</i>	
<i>Method</i>	<i>Description</i>
GetTransportDomain	Get the current transport domain
GetTransportPath	Get the current transport path
PickTransports	Allows a user to pick a transport from /usr/sap/trans

5.11 ZCL_DT_UTILITIES

This class is a collection of useful methods. In practice the class will need to be re-defined as the methods 'FindBackendRFCDest' and 'FindBackendSAPServer' will depend on your own system landscape.

<i>Instance class</i>	
Method	Description
FindBackendRFCDest	Find the back-end RFC destination if this system relies on R/3
FindBackendSAPServer	Find the SID of the back-end SAP system if this system relies on R/3
GenerateGUID	Generate a 32-bit GUID
IsDebuggingEnabled	Has the user specified debugging by setting global parameter ZDT_DEBUG in their user profile. This allows the programmer to add additional debug code to their program and enable it without coding change or setting manual break-points.
IsDynpro	Determines if this call is called form within the SAPGUI or from an non SAPGUI environment e.g. RFC or web service.
PickClient	Dropdown picker of all the available clients within the current SAP system
pickRFCDestination	Dropdown picker of all the available SAP RFC destinations within the current SAP system

5.12 ZCL_DT_XML

Convert an ABAP data structure/table to XML or additionally to HTML if used in conjunction with an XSLT program. This class is especially useful if you wish to convert data into a format suitable for displaying to the end user.

<i>Static class</i>	
<i>Method</i>	<i>Description</i>
ConvertStructureToHTML	Converts a data dictionary structure to HTML using an XSLT program
ConvertStructureToXML	Converts a data dictionary structure to XML
TransformViaXSLT	Transform XML into HTML using an XSTL program.

6 Table Maintenance

The following tables have table maintenances associated with them.

Table	Description
ZDT_FTPHOSTS	FTP hostname and description
ZDT_FTPPATHS	FTP hostname and file path mapping
ZDTC_FTPVARDET	FTP variant mapping to include server, path, ftp method, username and password
ZDTC_FTPVARIANT	FTP Variants