

Supplement #30

USA: +1 603 924 8818 • techsupport@softlanding.com • <https://support.softlanding.com>

EMEA: +44 1293 872101 • softlandingsupport@macro4.com • <https://support.softlanding.com>

TURNOVER® FOR ISERIES EXITS AND APIS

From the beginning, TURNOVER® has been developed to provide a framework for development. This means TURNOVER® for iSeries not only has to support your development, from beginning to end, but it also must work with dozens of products such as Hawkeye's Pathfinder™, or tools like LANSA™ and CA 2E™, and it must be able to manage applications like JD Edwards® World, and BPCS®. This has spawned dozens of Exits and APIs, born of the necessity of enabling various products to work with TURNOVER® for iSeries. These APIs are integral to TURNOVER® for iSeries. In fact, for ease of maintenance, our developers have used many TURNOVER® for iSeries APIs in developing TURNOVER® for iSeries. Most of the APIs available are programs encapsulated in commands. Most have help text that explains its function and defines its command parameters.

The Exits are points throughout TURNOVER® for iSeries that look for user-written programs and run them, if they are specified. To simplify management of Exits, we have provided you with an Exit panel (see Main Menu option 8, submenu option 7.) Here you can view all of the exit points, see help text related to each, and browse or copy sample source we have provided as a starting point. (The exit points are briefly described on the following pages; see your product online help for complete details.) You specify your exit and activate it on this panel:

6/28/01 14:55:56		Work with Exits		Your Company, Inc. YOURSYS	
5=Browse exit program		7=Work with sample exits		8=Help	
Exit	Description	Active	Program	Library	Library List
EXIT 1	Form Finalization Processing	N			*CURRENT
EXIT 2	Pre-Distribution Form Exit	N			
EXIT 3	Pre-Distribution Library Exit	N	TOEXIT3	SMITH	
EXIT 4	Receive of Distribution	N			
EXIT 5	Determining Source file names	Y		*LIBL	
EXIT 8	Form Initialization phase	N			*CURRENT
EXIT 9	Post-Checkout Exit	N	TOEXIT9	SMITH	
EXIT 10	Manual check-in	N			
EXIT 11	Form Completion Processing	N			*APPL
EXIT 12	Add a line to TurnOver form	N			
EXIT 13	Pre-run Error check Exit	N			
EXIT 14	Pre-Checkout Processing	N	TOEXIT14	SMITH	
EXIT 15	Add PWL Item	N			
EXIT 16	Form Approval	N			
EXIT 17	Document pre-checkout exit	N			
					More...
F3=Exit F8=IFS Exits F12=Cancel F21=Command line					

Caution: Be aware that an active exit program impacts all changes managed by TURNOVER® for iSeries. Therefore, they should be coded very carefully, and target just those applications and levels you intend. Test them carefully.

TURNOVER® for iSeries Exits and APIs

Exits and APIs define the borders of TURNOVER® for iSeries. Anything on the inside is the responsibility of UNICOM Systems, Inc.; anything on the outside is yours. It is that simple. From a proprietary point of view, you own the exits and programs that incorporate our APIs, we own everything else. If you stray inside the boundary and call other TURNOVER® for iSeries programs or use TURNOVER® for iSeries files, and then we make a change, we make no apologies. That is our territory and we can do anything we want there. However, we will avoid changing Exits and APIs, unless, of course, we find an error and need to correct it. This is important. You can write to our Exits and APIs, comfortable in the knowledge that the most you may need to do is recompile programs, if we change a file on which a program is based. So, generally, you can be sure that Exits or programs that use APIs will not need to be changed when we come out with a new release. If you need to change them, we will let you know in our Release Notes.

Caution: If you have trouble understanding what a particular Exit or API does, call us. We would rather you ask before you code, rather than try to straighten out a problem after the fact!

EXITS

The following is a list of exit points available in the TURNOVER® for iSeries product. Your TURNOVER® for iSeries product's online help text contains the information you need to write an exit program; refer there for specific details (such as parameter lists).

EXIT 1 Form Finalization Processing Exit

Exit 1 is called during the final stages of the TURNOVER® for iSeries form run. If you have a sequence of commands or a task that you want executed every time you have run a form, Exit 1 is the appropriate place to perform these commands.

EXIT 2 Pre-Distribution Form Exit

After a form has run successfully on your development computer and the distribution job has been initiated, Exit 2 is called. The exit is executed when the objects have been moved into the distribution library, but the library has not yet been sent to the target system(s). Exit 2 is called once for each form being distributed.

Note: The TOEXIT02B sample exit program is available for monitoring form confirmation. This program messages the distributing computer if a distributed form does not confirm after some interval.

EXIT 3 Pre-Distribution Library Exit

Exit 3 is called as part of the distribution job. It is called once per distribution job and after the following: the distribution library has been created (Thhmmssjjj); the temporary work library has been created (TnnnnnnnS); and the objects have been moved into the distribution library. Exit 3 can be used to move additional objects or other pertinent information into the distribution library before sending it to the target system(s).

EXIT 4 Receive of Distribution Exit

Exit 4 is called by the form receive job on the target system(s). If you have used Exit 3 to load the distribution library with information necessary for running the form(s) on the remote computer, you can use Exit 4 to unpack this information and move it to the desired location.

EXIT 5 Determining Source File Names Exit

Exit 5 is called whenever TURNOVER® for iSeries needs to determine a source file name, such as during checkout and when adding or changing a line on a form. If your application has source for one type code that resides in more than one source file, you will need to make use of this exit. Exit 5 is controlled within the TURNOVER® for iSeries global defaults, Source file name exit program and Library name (Main Menu option **8**, followed by option **5**).

EXIT 8 Form Initialization Phase Exit

If you have a sequence of customized commands or a task that you want executed just before every form run, Exit 8 is the appropriate place to do this work.

EXIT 9 Post-Checkout Exit

Exit 9 is called after a source member, object, or message ID has been checked out and copied to your development area. This exit can come in handy when working with source members or objects that are associated with supplemental files such as control files or help files.

EXIT 10 Manual Check-in Exit

Exit 10 is called when you use one of the manual check-in options for source code on the Checkout/Check-in Menu. Use this exit to perform additional check-in security processing, and/or other custom processing.

EXIT 11 Form Completion Processing Exit

Exit 11 is called during the final stages of a TURNOVER® for iSeries form run, after sending the form completion message(s). When a form has been run, completion messages are automatically sent to both the form programmer and the application administrator. Once a form runs and the completion messages are sent, you can use Exit 11 to propagate messages to users on other computer systems.

EXIT 12 Add a Line to TURNOVER® for iSeries Form Exit

Exit 12 is called whenever a line is added to a form, you might decide to change or evaluate one or more of the many parameters passed in to the exit program.

EXIT 13 **Pre-Run Error Check Exit**

Exit 13 is called during the form pre-run error checking process, and lets you to perform error checking beyond that which is automatically invoked by TURNOVER® for iSeries. (For a complete list of error checks that TURNOVER® for iSeries makes, see *Chapter 7* in the *TURNOVER® for iSeries User Guide*.)

EXIT 14 **Pre-Checkout Processing Exit**

Exit 14 is called before source members/objects are checked out and the source code copied to your development area. Given the list of parameters that are passed to the exit program, there are many different conditions you can evaluate to determine whether or not to allow a checkout request to proceed.

EXIT 15 **Add Programmer Worklist Item Exit**

When you indicate that you want to add an item to a worklist, the CL program ADDPWLITM is called. This program checks data area TOEXIT15 for the name of a user exit program. If it exists, the Exit 15 program is called before adding any items to a worklist. Within your Exit 15 program, you might choose to modify any of the passed parameters before adding an item to your worklist.

EXIT 16 **Form Approval Exit**

Whenever the form approval process starts in TURNOVER® for iSeries, your Exit 16 program is run. Exit 16 can be used to further condition form approval of the current application or some other parameters.

EXIT 17 **Document Pre-Checkout Exit**

Exit 17 is called before a document has been checked out and the document copied to your development folder. Perhaps you want to allow specific documents to be copied only to a project development folder, as opposed to the programmer's development folder. You can include these kinds of evaluations in your Exit 17 program.

EXIT 18 **Document Post-Checkout Exit**

Exit 18 is called after a document has been checked out and copied to your development area. You might write an Exit 18 program to raise a flag or send a message when company sensitive documents are checked out.

EXIT 19 Pre-Form Group Exit

Exit 19 is called before processing a form group. You might use this exit to check through the forms in the form group, check some condition, and either stop or continue processing, based on the result of the condition.

EXIT 20 Post Form Group Exit

Exit 20 is called after form status is updated; Exit 1 is called beforehand. If you want to perform some processing that depends on form status, use this exit rather than Exit 1.

EXIT 21 Post-Receive Exit

Exit 21 is called by the form receive job on the remote production computer. If you have used Exit 3 to load the distribution library with information necessary for running the form(s) on the target system(s), you can use Exit 21 to unpack this information and move it to the desired location.

EXIT 22 Post Add-Line Exit

Exit 22 is called after a line is added to a form. This exit might be used if you need to update a database to reflect the form and line number associated with the object being added to the form.

EXIT 23 Create Change Impact File Exit

Exit 23 is called after a Synchronizer change impact file is created. For example, you can use this exit to send warning messages based on programs that are listed in the change impact file being used to merge the vendor's changes.

EXIT 26 Related Forms Creation Exit

Exit 26 is called during the form creation process if there are applications related to the application for which you are creating the form. For example, you might use Exit 26 to create related forms only for selected application levels.

EXIT 27 Remote Configuration Wizard Pre-Save Exit

Exit 27 is called during the Remote Configuration Wizard, just before saving the work library to a save file. One way to use this exit is for adding additional objects to the work library to be saved and sent to the remote. For example, maybe all of the libraries for the application are saved into save files and included here. You can then use Exit 28 on the remote system to extract those libraries. This could be a way to automate literally sending an entire application to a new system.

Another reason to use this exit is to prevent the Remote Configuration Wizard from running during normal business hours. You can use this exit to end the process.

EXIT 28 Remote Configuration Wizard Post-Receive Exit

Exit 28 is called during the receive process of the Remote Configuration Wizard, just after restoring the save file to the work library, but before any application or system definitions are restored. For example, you might use this exit to restore additional objects you sent using Exit 27, or to change the library names contained within the applications you sent.

EXIT 29 Form Confirmation Exit

Exit 29 is called whenever confirmation is received for any of these form functions:

*CREATE	Form created.
*COPY	Copied to form nnnnnnn.
*APPROVE	Form approved.
*SBMERR	SBMJOB error.
*SUBMIT	Form submitted.
*RUNNING	Form started running.
*RUN	Ran with status xxxxxxxx.
*DISTRIB	Distributed to xxxxxxxx.
*RECEIVE	Form received.
*CANCEL	Canceled by xxxxxxxxxx.
*RESET	Form reset by xxxxxxxxxx.

You might use this exit if you want to require a successful automatic distribution to one remote computer before a form can be distributed to other remote computers.

Note: The TOMSGRCV confirmation job must be running on your remote computer in order for this exit to be called. The TURCRARE job is NOT among those set up automatically in subsystem TSERVER when you install TURNOVER® for iSeries. For information about setting up the TURCRARE form receive job on your remote computer, see TURNOVER® for iSeries Supplement #25.

EXIT 30 Conflict Resolution Exit

Exit 30 is called when you attempt to resolve a related application conflict. You can use this exit to enforce controls over who can resolve conflicts and what resolution codes can be used. You can also use this exit to perform additional processing when a conflict is resolved.

EXIT 31 Lock Check Exit

Exit 31 is called when TURNOVER® for iSeries form lock checking detects a lock on an object. You can use this exit to automatically remove locks on the object.

This exit can also be used when the TURCLOCK data area has been set to allocate files/members immediately after creation, instead of allocating the originals before replacing them. If the TURCLOCK data area contains a value of *X31 and this exit has been activated, then TURNOVER® for iSeries calls the exit program in place for this exit when it cannot allocate files or members immediately after creating them.

Note: If the **&LOCKED** parameter returns a value of **N**, TURNOVER® for iSeries proceeds as if the lock has been removed. If the lock has really NOT been removed, the form might recover.

Xn EXIT Create Command Replacement Variable Exit

This exit is called to resolve the "&X0", "&X1", "&X2", ... "&X9" replacement variables in creation commands (global type code definition).

PROJ 01 Timesheet Exit

This exit is called (for both the 5250 and Clients) during an interactive add, change, or delete of a timesheet entry. The purpose of this exit is to let you stop a timesheet entry from being added, changed, or deleted.

PROJ 02 Escalation Blackout Date/Time Exit

Project exit 2 is called in the escalation queue monitor job. It is called every time the job "wakes up" before it does any processing. Your exit program can then examine the current date/time and decide if the escalation monitor should continue its processing, or just go back to sleep.

PROJ 03 Escalation Blackout Item Exit

Project exit 3 is called in the escalation queue monitor job. It is called for each item that the job is about to process. It does not call the exit program for items whose wakeup date/time is still in the future. Your exit program can then examine the current item being processed and decide if processing should continue, the item should just be deleted from the queue, or the date/time for the item should be pushed into the future.

PROJ 04 Task Authorization Exit

Project exit 4 is called when the user takes the option to add, change or review a task. It is called from all support user interfaces (5250, Client/Server, Eclipse/WDSC, and EXPRESSDESK® for iSeries) so this should be taken into account when writing your exit program.

PROJ 05 Task Validation Exit

Project exit 5 is called before saving a task, after all built-in validations are performed. It is called from all supported user interfaces (5250, Client/Server, Eclipse/WDSC, and EXPRESSDESK® for iSeries) so this should be taken into account when writing your exit program.

XREF 01 Cross-Reference Exit 1: RPG Source Scan

X-Ref exit 1 is called during the source scan phase of the TURNOVER® for iSeries Cross-Reference process for an RPG program. This phase detects and writes cross-reference records for /COPY usage and externally defined data structures.

The purpose of this exit is to allow you to enhance this process by performing your own processing. You also have the option of telling TURNOVER® for iSeries to skip its processing.

APIS

Whereas Exits are called at various times while running TURNOVER® for iSeries to do additional processing, APIs let you to write your own programs to run various TURNOVER® for iSeries functions.

For example, you may want to write a program that retrieves a list of objects, creates a form and then adds those objects to the form. Or you may want to create a project task from information available in another system you are running. With APIs, you can accomplish these tasks without writing complex code to update TURNOVER® for iSeries' database or accomplish processing that is already done efficiently by TURNOVER® for iSeries.

APIs also free you from the impact of data base changes. Our programs access TURNOVER® for iSeries' database, not yours. We take responsibility for updating these if files change or underlying logic is revised. The process still looks the same to you. Normally, we do not change API parameters from release to release. In fact, we may create a new API simply to preserve the integrity of one you are using. We do not guarantee this, but you can be confident that changes to API parameters and function will not change unless absolutely necessary.

For ease of use, TURNOVER® for iSeries APIs are delivered in the form of commands, as described below.

General purpose APIs

ADDCMD

Add Command to Form

Is used to add a pre-run or a post-run command to a form.

ADDCMDI

Add Command to Form (*Interactive*)

This is the interactive version of the **ADDCMD** command that can be executed from a command line.

ADDFORM

Add TURNOVER® for iSeries Form for Application

A form can be added for a specific application. This command requires that all previously defined authorities and rules for the application be adhered to.

ADDFORMI

Add TURNOVER® for iSeries Form for Application (*Interactive*)

This is the interactive version of the **ADDFORM** command that can be executed from a command line.

ADDLINDLO Add Document Line to Form

Adds a line to a form that is a DOCUMENT type.

ADDLINEDLOI Add Document Line to Form (*Interactive*)

This is the interactive version of the **ADDLINDLO** command that can be executed from a command line.

ADDLINE Add a Line to Form

Allows a user to add a line to an existing form. You must have ADD authority for the application to execute this API.

ADDLINEI Add a Line to Form (*Interactive*)

This is the interactive version of the **ADDLINE** command that can be executed from a command line.

ADDUSRREF XREF Add User Reference

This command is used to populate the X-Ref table with user-specific cross- reference information, not already provided by TURNOVER® for iSeries.

ADDXRFTBL Add XREF Table

This command adds a member with the same name as specified in the application definition X-Ref Table parameter.

APVFORM Approve a Form

Used to approve or cancel a TURNOVER® for iSeries form. All form approval rules still are in effect when using the **APVFORM** command.

CHECKIN Check in Objects

Allows a user to manually check-in a source member or object. User must have check-in authority to the application and the item must already be checked out to execute this command.

CHECKOUT Check Out Objects

Allows an item to be checked out by object or by source member. The user must have all the proper authorities to use this command and all other checkout rules apply.

CHKCHGOBJ Check Changed Objects

Is used to approve changes made by other programmers as part of the form promotion process when multiple checkouts are enabled in TURNOVER® for iSeries.

CHKOUTDLO Check Out Document

Used to reserve a new document name and check it out.

CPYFORM Copy TURNOVER® for iSeries Form

Allows a user to copy one TURNOVER® for iSeries form to another. The parameters on this command allow a variety of options for customizing the form that is to be built through this process.

CPYFORMI Copy TURNOVER® for iSeries Form (*Interactive*)

This is the interactive version of the **CPYFORM** command that can be executed from a command line.

CRTRCVFRM Create Recovery Form

A recovery form (not a form in **RECOVER** status) allows a user to “undo” or reverse changes made by the form promotion process. The application must have been setup to archive objects in order to be able to create and run a recovery form.

CRTRCVFRMI Create Recovery Form (*Interactive*)

This is the interactive version of the **CRTRCVFRM** command that can be executed from a command line.

DBRFORM Check Database Relations and Add LFs to Form

This command is used to check the objects on a form for logical file dependencies. When necessary, LFs are added to the form as COPRs.

DLTSYNCHGF Delete Synchronizer Change Impact File Entry

When using Synchronizer, this command deletes a previously created change impact file entry.

DSTFORM Distribute Form

DSTFORM is used to distribute a form to one or more systems, or to add a form to a distribution job.

DSTFORMI Distribute Form (*Interactive*)

This is the interactive version of the **DSTFORM** command that can be executed from a command line.

LODOBJREF XREF Load Object Reference

This command loads the cross-reference data into the X-Ref Table for the object(s) specified. This can be done a single object or all objects in a library.

PRMDISFORM Promote Distributed Form

This command is functionally similar to the **PRMFORM** command, but this command is run on the remote (receiving) computer.

PRMFORM Promote a Form

This command will copy a designated form and if indicated, will run the form.

RCVFORM Receive a Form

RCVFORM is used to receive a form that has been distributed to a remote (receiving) computer.

RMVLINE Remove Line from Form

Use this command, either interactively or in a program, to remove a line from a TURNOVER® for iSeries form.

RMVOBJREF XREF Remove Object Reference

This command removes cross-reference information from the database for a particular object.

RMVUSRREF XREF Remove User Reference

Removes user-specific cross-reference information from the X-Ref Table.

RMVXRFTBL Remove XREF Table

This command removes specified members from the cross-reference files.

RUNFORM Run a Form

RUNFORM is used to run a form as long as the status of the form is either **READY** or **RECVD**.

RUNFORMGRP Run Form Group

RUNFORMGRP is the command used to run TURNOVER® for iSeries forms.

SEQFORM Resequence a Form

This command is used to reorder lines on a form according to type code. The order of objects is defined in the TURNOVER® for iSeries type code file.

TADDFORGRP Add Form to Form Group

This command adds a form to a previously created form group. You can use the **TCRTFORGRP** command to create a form group.

TCRTFORGRP Create Form Group

This command creates a form group. Once you create a form group, you can use the **TADDFORGRP** command (see previous) to add forms to the group.

TWRKCON Work with Conflicts

Work with conflicts between related applications.

TWRKLIC Work with Licenses

Check license information and apply product authorization codes.

Project system APIs

The following TURNOVER® for iSeries Projects system APIs are available to help TURNOVER® for iSeries interact with other products, such as Lotus Notes. (To see parameters for these commands, use **F4** to prompt.)

TADDREQ Add Requester

This command adds requesters to the TURNOVER® for iSeries Requester file.

TADDREQCMT Add Requester Comment

This command associates one line of free format text with a record in TURNOVER® for iSeries' Requester file. (To attach multiple lines of text, issue this command multiple times.)

TADDTSK Add Task

This command adds a task to a project you specify, and requires a project identifier and other task data. It returns the new task ID created by the TURNOVER® for iSeries. Field defaults can be derived from the task defaults of the target project.

TADDTSKDTL Add Task Details

This command adds one 79-character detail line to a project task. If you were adding details, you would call this command repeatedly for each line of task detail text. If you were updating task details, you would read all of the detail lines from the TURNOVER® for iSeries task database, update it in a PC-based text editor, and then write it all back using this API command. **TADDTSKDTL** includes a delete task details parameter so that you can delete the old details the first time through, and then write an entire new set of detail records. The rationale is that the editor you use might revise any part of the text, and line justification might also have changed it as well.

TADDTSKI Add Task (interactive)

Like **TADDTSK**, this command adds a task to a project you specify. It also requires a project identifier and other task data. However, it does not return a Task ID, and is intended to be run interactively, either from a

command line or from within a program. Field defaults can be derived from the task defaults of the target project.

TADDTSKLNK Add Task Link

Use this API in programs to add task, shortcut, document, requester, or URL links to TURNOVER® for iSeries tasks.

TASGTSK Assign Task

This command assigns a helpdesk task to a valid working project, and requires that you enter project, task and subtask identifiers for a helpdesk project as well as a working project identifier. It has a list of additional parameters that allow you to override other task fields, as appropriate. Field defaults are derived from the project defaults of the target project.

TCHGREQ Change Requester

This command allows you to change the information defined for a requester in the TURNOVER® for iSeries requester file.

TCHGAPPLIB Change Application Libraries

This command changes the libraries in an application definition.

TCHGTSK Change task

This command lets you change the contents of various fields in a task record. It requires you to enter project, task and subtask identifiers for the task you want to change. This command uses the special value ***SAME** for every parameter, so only values which need to be changed should be specified.

TCPYPRJ

Copy Project

This command duplicates a project definition, with the option of copying tasks and subtasks as well.

Note: This command does not create links between tasks in the original project and tasks (and subtasks) that are copied. See **TCPYTSK** below.

TCPYTSK

Copy Task

This command duplicates a task or subtask from one project to another. By default, this command creates a link between the original task and the new, copied task.

TCPYTSK2

Copy Task version 2

This command does the same work as **TCPYTSK**, but in batch. It returns the new task ID.

TDLTAPPDFN

Delete Application Definition

This command deletes an application definition, after you've deleted all existing levels for the application. If authority issues, or existing forms or checkout records for the level prevent you from deleting levels, you won't be able to delete the application definition.

TDLTAPPLEV

Delete Application Level

This command deletes a level from an application definition. To delete a level, you must have System Defaults authority to the level. Also, before you can delete a level, you must check in any items checked out from the level and purge any existing forms for the level, without retaining object history.

TDLTPRJ Delete Project

Removes a project from your system. (User authority settings apply.)

TDLTREQ Delete Requester

This command removes email or comment information from a requester record, OR removes an entire requester record, from TURNOVER® for iSeries' requester database.

TDLTTSK Delete Task

Removes a task from a project. (User authority settings apply.)

TMOVTSK Move Task

Transfers a task from one project to another.

TMOVTSK2 Move Task version 2

This command does the same work as **TMOVTSK**, but in batch. It returns the new task ID.

TPRJMSG Send Project Message

Sends a project message (must be defined in the Project Messages file) to an IBM i system user profile message queue.

TRTVREQ Retrieve Requester

This command returns information about a specific requester to your calling program.

TSNDDST Send OfficeVision Message

This command uses the IBM i **SNDDST** command to build and send a project message to a recipient you identify, including users that are using IBM i SMTP/POP3 electronic mail service.

TSNDMAPI Send MAPI Message

Constructs and transmits a project message to a recipient in a MAPI-compliant email system, including POP3.

TUPDHLPTSK Update Helpdesk Task

Maps Working task values to their corresponding Helpdesk task counterparts. For use in Escalation Table Entries. See online Help for more.

WDDYNUPD Initiate Wisedesk Dynamic Updating

Manages the system autostart job that monitors the Wisedesk data queue. For a complete description of this command, see the online Help.

Notes:

1. The only validity checking supported by these APIs pertains to passing a valid resource and valid status. The programs you code to call the APIs must supply validity checking on other fields.
2. Most commands will end in error if an error is encountered, such as insufficient authority to the function, so you should use a MONMSG with the API.

New APIs are created frequently, so if you do not see an API that you think you need, call us. Chances are, we will already have the one you need.

Truly yours,

Technical Support Staff