NetLink 기능리스트

상세설계서(기술명세)

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 로드밸런싱(LB) | 로그인할 서버접속 (SC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.LB.SC** | **기능번호** | **LO.LB.1** |
| **기능설명** | 서버에 접속하여 실제 접속할 서버 IP를 확인, 접속을 요청한다. | | |
| **입력** | 접속할 서버 IP / Port | | |
| **출력** | 실제 접속할 IP | | |
| \* Step 1: 입력 되어진 서버IP와 포트에 대해 입력값 벨리데이션을 한다.  \* Step 2: 소켓을 생성하여 망연계 VIP로 접속하여, 1002(접속IP 정보 가져오기) 메시지를 전송한다.  \* Step 3: 응답(2002)를 대기하고, 2002수신후 Real IP를 리턴한다.  \* Step 4: 2002응답이 50초 Timeout시간동안 응답이 수신되지않으면, 에러코드: 미응답으로 리턴한다 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 로드밸런싱(LB) | 서버설정정보 수신 및 로그인(RL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.LB.RL** | **기능번호** | **LO.LB.2** |
| **기능설명** | 에이전트는 서버에서 시스템구분정보를 수신하여 시스템타입에 맞는 로그인창을 기동, 로그인한다 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: 서버에서 받은 2005 응답 정보들로 Login창을 기동  \* Step 2: Login 창은 LoginType에 따라 필요한 정보를 입력받는 UI로 구성  \* Step 3: Login 창은 접속서버(내.외부)에 따라 다른 배경 이미지를 표현  \* Step 4: 사용자가 로그인 시도시 로그인처리 및 agent UI 구성에 필요한 정보들 수신 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 통신(COM) | SSL 접속 및 data 통신 (SSL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.COM.SSL** | **기능번호** | **LO.COM.1** |
| **기능설명** | 3435로 접속할때, SSL 접속 및 통신 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: SOCKET을 사용하여 Connect 시도  \* Step 2: 성공하면, TLSv1\_2\_method 부터 TLSv1\_method까지 내려오며, 서버와 SSL\_connect 시도  \* Step 3: 성공하면, SSL\_get\_verify\_result, SSL\_get\_peer\_certificate api 등으로 SSL\_connect 에 대해 확인  \* Step 4: 이후로 Server와 통신하는 모든 과정은 이 SSL을 통해서 진행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 통신(COM) | packet 암호화 & 압축 (PTENC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.COM.PTENC** | **기능번호** | **LO.COM.2** |
| **기능설명** | 3435로 통신하는 통신전문은 Tag, Data로 구성되며, Data는 AES256으로 암호화를 거침 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: $SEEDKEY 통신전문(통신전문의 Tag 값을 암호화 할지 유무, RSA비대칭 Key, Packet Body를 AES256 암호화 여부) Server로 전달  \* Step 2: $SEEDKEY\_ACK 에서 받은 SEEDKEY Tag의 값을 RSA private key로 복호화해서, session key로 사용한다.  \* Step 3: Server에서 내려주는 session Key를 서버와 주고받는 Packet의 Body Data, 통신전문의 Tag 값을 AES256 암.복호화 할 때 사용  \* Step 4: 송.수신하는 파일의 Data를 AES256 암.복호화 할 때에는 접속하는 사용자ID를 KEY로 사용  \* Step 5: 서버와의 접속이 끊어져서 Reconnect 하게 될 때에도 Step1과 동일하게 $SEEDKEY 통신으로 Server에서 session Key를 다시 받아 동작 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 통신(COM) | login, 통신과정 (LOGIN) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.COM.LOGIN** | **기능번호** | **LO.COM.3** |
| **기능설명** | Agent 실행후 최초 Server로 로그인때, 여러가지 통신 전문을 Server로 보내, 관련 정보를 얻고 UI 구성 및 동작을 설정 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: $SEEDKEY 로 서버와 암호화 통신에 사용할 AES256 Key를 얻음  \* Step 2: $SESSIONCOUNT 로 로그인 하려는 사용자 ID의 session 개수를 알아냄  \* Step 3: $Bind 로 서버에 인증, 각종 Server 정책들을 받아서 agent UI 및 동작 설정  \* Step 4: $USERINFOEX 로 로그인한 사용자 정보 조회(이름, 직위, 기타등등)  \* Step 5: $ApproveDefault 로 기본결재라인 정보를 조회  \* Step 6: $APPR\_INST\_CUR 로 로그인한 사용자가 지정한 대결재자 정보 조회  \* Step 7: $SYSTEMRUNENV 로 시스템의 환정정보 조회 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | SESSION정책(SP) | 동일계정 중복Login SESSION 정책(SP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.SP.SP** | **기능번호** | **LO.SP.1** |
| **기능설명** | 동일망, 동일계정으로 login 시도때,  강제처리 : 후에 로그인 agent가 유효, 기존 agent는 logOut 처리  선택처리 : 후에 로그인 agent가 기존 agent는 logOut 되더라도 login 진행할지, 아니면 login중단할지 선택 | | |
| **입력** |  | | |
| **출력** |  | | |
| 강제처리일때  \* Step 1: $SESSIONCOUNT 로 session 개수를 문의하지 않고 login 진행( $Bind 송신 )  \* Step 2: 이미 계정으로 로그인된 agent가 있다면, 이번 로그인때 $LOGOUT 전문이 서버에서 오며 agent에서는 message를 띄우고 ‘확인’을 누르면 agent는 종료  선택처리일때  \* Step 1: $SESSIONCOUNT 로 로그인 하려는 사용자 ID의 session 개수를 알아냄  \* Step 2: 결과가 0 보다 크면, 사용자에게 현재 다른PC에서 해당 계정으로 로그인 중인데, 계속 로그인을 진행할지를 물어봄.  \* Step 3: UI에서 ‘예’를 선택하면 지금 사용사의 agent login 진행, ‘아니오’선택하면 현재 agent 종료.  \* Step 4: 이미 계정으로 로그인된 agent가 있다면, 이번 로그인때 $LOGOUT 전문이 서버에서 오며 agent에서는 message를 띄우고 ‘확인’을 누르면 agent는 종료 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 접속서버 사용자지정(US) | 전송통제서버 IP 입력(IP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.US.IP** | **기능번호** | **LO.US.IP** |
| **기능설명** | 접속 서버 IP를 처음실행때 입력받아 이후에 거기로 접속(POC에서만 사용) | | |
| **입력** | 접속하고자 하는 서버IP | | |
| **출력** |  | | |
| POC 때에만 사용되는 기능  \* Step 1: agent 설치후 첫실행때, 접속할 IP를 입력하는 UI 출력  \* Step 2: 접속하고 하는 서버의 IP 입력  \* Step 3: 입력한 IP의 3435 Port로 접속이 가능하면, 해당 IP 암호화 하여 SecureGate.ini 파일에 AES256으로 암호화 하여 저장, Key는 고정값  \* Step 4: 입력한 IP의 3435 Port로 접속이 불가능하면, error Message 출력후 Step 2 로 이동  \* Step 5: agent가 SecureGate.ini 파일에 AES256으로 암호화된 IP주소 값 복호화하여 서버접속에 사용  \* Step 6: agent가 접속할 IP를 변경하고 싶을 때, agent 메뉴> 시스템(S) > 전송통제서버 IP 변경 선택  \* Step 7: 접속하고자 하는 IP를 입력, 저장 후 agent 를 종료 및 재실행하게 되면 변경한 Server로 접속 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | ID/PW 인증 - 자동로그인(IDPW) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.IDPW** | **기능번호** | **LO.UA.1** |
| **기능설명** | 사용자가 입력한 계정과 비밀번호를 서버에 전송하여 사용자 인증을 수행한다.  자동로그인 체크 설정시 로그인 성공한 이후부터는 ID/PW 입력없이 자동로그인된다. | | |
| **입력** | 접속하고자 하는 계정(ID), password | | |
| **출력** |  | | |
| \* Step 1: Login창에서 사용자 계정과 비밀번호를 입력하고, login 실행  \* Step 2: 잘못된 ID나 PASSWORD 입력시에 “사용자 인증에 실패 하였습니다.”라는 메시지 출력  \* Step 3: $BIND\_ACK 의 Result 값, 1~8 까지 login 실패사유  1 : 사용자 인증실패  2 : 중복로그인  3 : 사용자 계정 일시 잠김  4 : 사용자 계정 영구 잠김  5 : 등록 되지 않은 IP/MAC정보  6 : 계정 정지  7 : 계정 기간 만료  8 : 파일 무결성오류 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | AD 인증(AD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.AD** | **기능번호** | **LO.UA.2** |
| **기능설명** | AD 서버 계정정보로 사용자 인증 수행, login 화면없음 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: AD 인증은 사용자 Login창 UI가 없음  \* Step 2: 현재 OS에 로그인한 계정을 ID, 현재 PC가 속한 domain을 대문자화 해서 password로 사용  \* Step 3: 실제 AD Server와 어떠한 인증절차를 거치지 않음  \* Step 4: 망연계 Server에 인증을 거칠 때, ADALDAPSVR 에 있는 domain 값과 다를 경우 인증실패 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | LDAP 연동(LDAP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.LDAP** | **기능번호** | **LO.UA.3** |
| **기능설명** | LDAP 을 사용한 사용자 인증 수행 | | |
| **입력** | ID, PassWord | | |
| **출력** |  | | |
| \* Step 1: SGConf.xml 파일을 읽음  \* Step 2: xml Tag 들을 Parsing 해서 메모리에 저장  \* Step 3: ID / PW때 사용했던 Login창과 동일한 Login창에 ID / PW를 입력 / 로그인 시도  \* Step 4: xml Tag 에서 <LDAP> 에 있는 서버 주소, Port(default:389)로 ldap\_open, ldap\_set\_option api를 사용하여 접속  \* Step 5: SGConf.xml 파일에서 <LDAP> 에 있는 인증용 Data를 가공  \* Step 6: Ldap Server에 ldap\_simple\_bind\_s Api로 Bind 시도  \* Step 7: Bind에 실패하면, "LDAP 인증에 실패 하였습니다."라는 message 출력  \* Step 8: Bind에 성공하면, 망연계 Server에 입력받은 ID / PW 정보로 Login 실행  \* Step 9: Ldap Login 때는 password를 체크하지 않도록 $sessioncount 전문 전송때, PasswordType 값을 9로 설정해서 패스워드 체크 안하도록 처리  \* Step 10: 망연계 Server로 Bind 성공후 기본적인 Login Process 동작 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | AD\_LDAP 연동(ALDAP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.ALDAP** | **기능번호** | **LO.UA.4** |
| **기능설명** | AD 서버 계정으로 Ldap 서버에 사용자 인증 수행 | | |
| **입력** | PassWord | | |
| **출력** |  | | |
| \* Step 1: SGConf.xml 파일을 읽음  \* Step 2: xml Tag 들을 Parsing 해서 메모리에 저장  \* Step 3: ID / PW때 사용했던 Login창과 동일한 Login창에 PW를 입력 / 로그인 시도, ID는 AD인증때 거친 현재 PC 로그인한 계정ID가 입력되어 있고, 사용자가 수정 불가  \* Step 4: pc가 속한 AD Domain이 xml Tag 에서 <AD> 에 있는 서버와 동일한지 확인  \* Step 5: xml Tag 에서 <LDAP> 에 있는 서버 주소, Port(default:389)로 ldap\_open, ldap\_set\_option api를 사용하여 접속  \* Step 6: SGConf.xml 파일에서 <LDAP> 에 있는 인증용 Data를 가공  \* Step 7: Ldap Server에 ldap\_simple\_bind\_s Api로 Bind 시도  \* Step 8: Bind에 실패하면, "LDAP 인증에 실패 하였습니다."라는 message 출력  \* Step 9: Bind에 성공하면, 망연계 Server에 입력받은 ID / PW 정보로 Login 실행  \* Step 10: Ldap Login 때는 password를 체크하지 않도록 $sessioncount 전문 전송때, PasswordType 값을 9로 설정해서 패스워드 체크 안하도록 처리  \* Step 11: 망연계 Server로 Bind 성공후 기본적인 Login Process 동작들 진행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | OTP 연동(OTP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.OTP** | **기능번호** | **LO.UA.5** |
| **기능설명** | 아이디 및 OTP번호를 사용한 사용자 인증 수행 | | |
| **입력** | ID, OTP번호 | | |
| **출력** |  | | |
| 내.외부망 agent중 하나만 OTP 번호를 사용한 인증을 할 수 있음  \* Step 1: 내부망 Server 설정, OTPUSE 값 1로 설정  \* Step 2: 내부망에 agent Login 수행  \* Step 3: OTP 생성 실행 - agent에서 6자리의 OTP 번호를 생성  \* Step 4: $makeOTP 전문으로 생성한 OTP 인증번호를 서버에 전송  \* Step 5: 3분간만 해당번호가 유효함을 UI 상에 표시  \* Step 6: 외부망 Server 설정, LOGINTYPE 값 OTP로 설정  \* Step 7: 외부망 agent 실행, ID / 내부망 agent에서 보여주는 OTP값 입력 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | PW\_OTP 연동(POTP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.POTP** | **기능번호** | **LO.UA.6** |
| **기능설명** | 패스워드 및 OTP번호를 사용한 사용자 인증 수행 | | |
| **입력** | ID, Password, OTP | | |
| **출력** |  | | |
| 내.외부망 agent중 하나만 이 Password와 OTP 를 사용하는 인증을 사용 할 수 있음  \* Step 1: 내부망 Server 설정, OTPUSE 값 1로 설정  \* Step 2: 내부망에 agent Login 수행  \* Step 3: OTP 생성 실행 - agent에서 6자리의 OTP 번호를 생성  \* Step 4: $makeOTP 전문으로 생성한 OTP 인증번호를 서버에 전송  \* Step 5: 3분간만 해당번호가 유효함을 UI 상에 표시  \* Step 6: 외부망 Server 설정, LOGINTYPE 값 PW\_OTP로 설정  \* Step 7: 외부망 agent 실행, ID / Password / 내부망 agent에서 보여주는 OTP값 입력  \* Step 6: Password / Otp 번호중 어느 하나라도 맞지 않으면 login 실패 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | NAC 연동(NAC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.NAC** | **기능번호** | **LO.UA.7** |
| **기능설명** | NAC에서 받은 ID를 사용해서 Login | | |
| **입력** | ID | | |
| **출력** |  | | |
| 지니NAC  \* Step 1: 우리 agent 실행 파일을 " authid: (아이디) “ 와 같은 parameter와 함께 실행해줌  \* Step 2: agent에서 해당 parameter와 함께 실행되었는지 확인  \* Step 3: string Parsing을 통해 ID값 추출  \* Step 4: 추출된 ID를 Password로도 사용  \* Step 5: $SESSIONCOUNT 를 보낼 때, 패스워드체크안함 으로 설정  \* Step 6: login 창 없이 로그인 동작 수행  \* Step 7: NAC 연동에 의해 실행된 agent가 Patch를 진행해야 될 때, NAC으로 실행되었고 어떤 ID를 전달 받았는지 값을 SECUREGATE.ini 파일에 저장해둠  \* Step 8: Updater Process에 의해 agent가 다시 실행되었을 때, SECUREGATE.ini 파일에 해당 내용이 있으면 NAC으로 실행된 것으로 판단하고 해당 값을 사용해서 agent NAC login 수행  NetMan Nac  \* Step 1: agent 실행되면, 특정 Registry 위치에 암호화된 NacID 값을 확인  \* Step 2: AES256 + Base64 로 인코딩 된 값을 해당 업체에서 알려주는 Key와 IV 값으로 해제  \* Step 3: 추출된 NacID를 agent login에 사용할 ID / Password로 사용  \* Step 4: $SESSIONCOUNT 를 보낼 때, 패스워드체크안함 으로 설정  \* Step 5: login 창 없이 로그인 동작 수행  TGATE Nac  \* Step 1: agent 실행되면, 특정 Registry 위치(KEY\_LOCAL\_MACHINE\Software\\Medialand\\Tgate\\Agent)를 찾음  \* Step 2: 값(InstalledPath)을 읽어 NAC 모듈이 설치된 위치를 알아냄  \* Step 3: TGATE에서 제공한 dll, exe 가 있는지 확인  \* Step 4: dll 에서 제공하는 api 를 사용해서 login에 사용된 ID / Password를 알아냄  \* Step 5: 추출된 ID / Password를 agent login에 사용  \* Step 6: $SESSIONCOUNT 를 보낼 때, 패스워드체크안함 으로 설정  \* Step 7: login 창 없이 로그인 동작 수행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | SSO 연동(SSO) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.SSO** | **기능번호** | **LO.UA.8** |
| **기능설명** | SSO에서 받은 계정정보를 이용해서 Login | | |
| **입력** |  | | |
| **출력** | ID, PW | | |
| SSO 업체가 너무 다양하고 대표적인 연동 업체 1곳을 정하기 어려우므로 몇사례만 언급  Dll api / web 접근방식 (RAON secure)  \* Step 1: SSO 업체에서 제공해준 dll 의 api를 사용  \* Step 2: 지정해준 web page, Port로 접속해서 sso token을 받아냄  \* Step 3: sso token으로 받은 값을 id / password로 사용  \* Step 4: $SESSIONCOUNT 를 보낼 때, 패스워드체크안함 으로 설정  \* Step 5: login 창 없이 로그인 동작 수행  Dll api 제공 방식  \* Step 1: SSO 업체에서 제공해준 dll 의 api를 사용해 id / password 를 얻어내어 agent login에 사용  \* Step 2: $SESSIONCOUNT 를 보낼 때, 패스워드체크안함 으로 설정  \* Step 3: login 창 없이 로그인 동작 수행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | GPKI(GPKI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.GPKI** | **기능번호** | **LO.UA.9** |
| **기능설명** | GPKI 인증서 파일을 선택하고 인증서 비밀번호를 입력하여 로그인을 수행 | | |
| **입력** |  | | |
| **출력** |  | | |
| GPKI 인증을 사용하기 위한 사전 작업   * GPKI 인증서버 설정   \* Step 1: agent에서 $GPKI\_RANDOM 으로 GPKI Random 키 서버에 요청  \* Step 2: GPKI Random 키 응답 받음  \* Step 3: GPKI Random 키로 인증서 서명 후 인증서 (bianary data) Server로 전달  \* Step 4: Server에서 통합인증서버(행안망에 있는 정부서버)로 인증서 검증 요청  \* Step 5: 인증서 검증 결과 응답을 받음  \* Step 6: agent로 인증서 검증 결과 응답과 사용자 ID 전달  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10:  \* Step 11:  \*. 내용 추가 필요 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | Google-OTP(GOTP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.GOTP** | **기능번호** | **LO.UA.10** |
| **기능설명** | smart Phone의 Google-OTP App을 활용, Google-OTP 등록 및 인증 | | |
| **입력** |  | | |
| **출력** |  | | |
| Google OTP 인증을 위한 사전 설정 필요   * 사용자별 인증방식 DB가 아닌 LDAP으로 지정 * 망연계WAS 필요   \* Step 1: agent login창에 입력한 ID / PW를 망연계WAS 통해 인증받음  \* Step 2: 인증성공시 Google OTP 등록정보를 받음  \* Step 3: Google OTP 번호입력 망연계WAS에서 인증성공여부 결과 받음  \* Step 4: 망연계 서버에 로그인 수행  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10:  \* Step 11:  \*. 내용 추가 필요 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 사용자 인증(UA) | CustomURI(CURI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UA.CURI** | **기능번호** | **LO.UA.11** |
| **기능설명** | CustomURI 를 통해 웹 포털의 계정과 연동하여 자동로그인 | | |
| **입력** | ID, PW | | |
| **출력** |  | | |
| \* Step 1: agent 설치때, CustomURI 에 의해 agent가 실행될 수 있도록 Registry에 관련 정보 등록  [HKEY\_CLASSES\_ROOT\securegate]  "URL protocol"=""  [HKEY\_CLASSES\_ROOT\securegate\shell]  [HKEY\_CLASSES\_ROOT\securegate\shell\open]  [HKEY\_CLASSES\_ROOT\securegate\shell\open\command]  @="\"C:\\HANSSAK\\SecureGate\\SGStart.exe\" \"%1\""  [HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\ProtocolExecute]  [HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\ProtocolExecute\securegate]  "WarnOnOpen"=dword:00000000  \* Step 2: 등록된 CustomURI 통해, 고객 Potal에서 "securegate://(ID):(Password)/" 방식으로 호출가능  \* Step 3: Registry에 등록된 SGStart.exe 실행파일에 parameter로 (ID):(Password) 가 전달되며, 실행  \* Step 4: SGStart.exe 는 관련 정보를 암호화 저장하고, SecureGate.exe를 실행  \* Step 5: SecureGate.exe 는 SGStart.exe가 저장한 암호화된 ID / Password 값을 복호화하여 Login 진행  \* Step 6: login 창은 표현되지 않음.  \* Step 7: web Portal에서 CustomURI를 통해 자료연계 agent를 바로 실행시킴 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 로그인(LO) | 자동로그인(UAL) | 자동로그인 / 해제(UAL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **LO.UAL.UAL** | **기능번호** | **LO.UAL.1** |
| **기능설명** | ID/PW 입력없이 자동로그인, 로컬에 저장된 ID/PW 값을 복호화하여 사용  메뉴 > 시스템 > 자동로그인해제 에서 자동로그인 해제가능 | | |
| **입력** |  | | |
| **출력** |  | | |
| 자동로그인 : 사용자의 계정정보 입력없이 agent가 Login 동작을 수행  PC에 계정정보를 암호화하여 저장하는 방식의 자동로그인 : ID/PW, LDAP  외부연동등으로 계정정보 저장없는 자동로그인 : AD, NAC, SSO 등  이중에 ID/PW, LDAP 에 대한 자동로그인만 해당  \* Step 1: login 창에서 ID / PW 입력, 자동로그인 체크후 로그인성공  \* Step 2: ID / PW 를 자체암호화 + Base64를 적용해서 SECUREGATE.ini 에 저장  \* Step 3: 해제를 원할때, agent 메뉴 > 시스템(S) > 자동로그인해제 를 실행  \* Step 4: SECUREGATE.ini 에 저장된 ID / PW 값을 지움  \* Step 5: ID 는 “UID=#####”, Password는 “UPW=#####” 형식으로 저장됨 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일보내기(FT) | 암호화(ENC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FT.ENC** | **기능번호** | **FT.FT.1** |
| **기능설명** | 전송파일 암.복호화 | | |
| **입력** |  | | |
| **출력** |  | | |
| 송신하려는 파일들을 agent는 1개의 hsz 파일로 만들어 보내는데, 이때 설정에 따라 암호화를 하며, 주로 AES256 암호화 방식을 사용.(Seed 암호화 혹은 무암호화로 설정가능)  \* Step 1: 파일의 data 암호화는 설정한 방식으로 암호화 하지만, 이때 사용하는 key는 별도의 암호화 과정을 거침  \* Step 2: HSZFRAMESLOT의 Slot에 전송할 파일들의 Data를 암호화해서 저장  \* Step 3: 암호화된 hsz 파일을 수신하는 쪽에서는 서버에서 주는 FileKey를 사용해서 복호화한다.  \* Step 4: 파일 미리보기때 평문 파일을 보내주는 server도 이 FileKey를 사용해서 복호화한다.  FileKey를 암.복호화하는 과정  \* Step 1: 사용자ID를 HsGenerateKey 통해서 자체암호화  \* Step 2: 이 암호화된 값을 다시 HsAes256EncryptKey에서 다시 암호화  \* Step 3: 이때 사용되는 key값이 “묻지마@문열어”인데 이 값도 평문으로 저장되어 있지않고, agent 처음실행때, RSA 공개키로 암호화해서 저장해놓음.  \* Step 4: FileKey 값을 구하기 위해 암호화 해제할때에만 RSA 비밀키로 복호화해서 잠시 사용함. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일보내기(FT) | hsz 파일 생성(HSZ) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FT.HSZ** | **기능번호** | **FT.FT.2** |
| **기능설명** | 송신할 파일 / 폴더 정보, 파일을 1개의 file로 만든 hsz 파일을 생성 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: 압축 및 파일 처리에 사용할 메모리 3.3MB 할당  \* Step 2: Server에서 전달 받은 HSZDEFAULTOPTION 설정값 확인  \* Step 3: HSZDEFAULTOPTION 설정값이 저장된 HszHead data를 hsz 파일에 저장  \* Step 4: 전송하려는 File들 전체크기 최대전송크기 넘는지 확인  \* Step 5: 장애발생시킬 수 있는 특수문자들 치완, 이후 utf8로 변환  \* Step 6: HszFrame64에 원본파일크기, 파일 / 폴더 / 공인인증서 구분정보, 시간정보 등을 넣어 hsz 파일에 저장  \* Step 7: HszFrameSlot 과 파일의 Data를 압축, 암호화한 Data를 hsz에 저장  \* Step 8: hsz 파일 생성 중간에 LinkCheck 서버로 전송  \* Step 9: 파일 1개의 data를 hsz 파일에 넣고난후에 HszFrame64 한번더 저장  \* Step 10: HszHead.frameCount 에 파일개수 정보 저장  \* Step 11: hsz 파일을 복호화 하는데 필요한 FileKey, hsz 파일경로 명과 함께 통신단에 전달 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일보내기(FT) | 무압축/무암호화(NENC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FT.NENC** | **기능번호** | **FT.FT.3** |
| **기능설명** | 서버 설정값에 따라 압축 및 암호화 단계를 생략 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: Server에서 전달 받은 HSZDEFAULTOPTION 설정값 확인  \* Step 2: 설정값은 16진수(ex. 0x00000010), 2진수화 진행  \* Step 3: 2진수 값에서 제일 오른쪽 값은 UTF8(1) / EUCKR(0) 인지를 결정  \* Step 4: 2진수 값에서 오른쪽 2째값은 32bit(0) / 64bit(1) 인지를 결정  \* Step 5: 2진수 값에서 오른쪽 3째값은 압축(0) / 무압축(1) 인지를 결정, 대상은 hsz 파일에만 적용. hsz을 생성 / 해제할 때 Z압축/해제를 사용할지 말지만 결정, 통신하는 전문 압축에는 영향주지 않는다.  \* Step 6: 2진수 값에서 오른쪽 4째값은 암호화(0) / 무암호화(1) 인지를 결정, 대상은 hsz 파일생성때 저장하는 파일Data, 파일Data 저장후 저장하는 HszFrame64 에만 적용, 통신하는 전문 암호화에는 영향주지 않는다.  \* Step 7: 2진수 값에서 오른쪽 5째값은 OS의길파일명 최대치를 미지원(0) / 지원(1) 인지를 결정  \* Step 8: agent는 UTF, 64bit 사용을 기본적으로 하도록 되어 있고, hsz 파일에 이 2가지 Option은 넣지 않는다. 넣으면, 해당 hsz 파일 압축해제시 error 발생한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일보내기(FT) | 송신 과정(SEND) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FT.SEND** | **기능번호** | **FT.FT.4** |
| **기능설명** | 전체진행 상황, 개별 진행 정보 표현, 지정한 Type으로 파일 서버에 전달 | | |
| **입력** | 송신하고자 하는 hsz 파일 | | |
| **출력** |  | | |
| \* Step 1: 압축으로 생성된 hsz 파일의 hash 값 구함  \* Step 2: Server의 PCCHECKSUM 설정값이 “FILECHECKSUM” 설정값으로 내려오는걸 확인  \* Step 3: 1:MD5, 2:SHA384, 3:SHA512, 해당하는 종류의 Hash를 구함  \* Step 4: ClientMem의 DownFileInfo64에 hsz 파일 및 기타 통신관련 정보들을 초기화 / 입력한다  (Workmode, totalpart, totalsize, curpart, sumsize, wfilename)  \* Step 5: 송신 진행정보를 표시하는데 사용하도록 hsz 파일 Size 정보를 UI 단에 전달  \* Step 6: $TR\_FILEUPLOAD 전문을 사용하여, hsz 파일의 내용을 연결된 Server로 전송  \* Step 7: ClientMem의 DownFileInfo64에 값들을 전송 반복횟수마다 최신정보로 갱신, 전송한 량을 UI단에 전달  \* Step 8: 파일 송신이 완료되면, UI단에 파일 전송 완료 정보를 전달  \* Step 9: $TRANSREQ 전문을 사용하여, 이번 파일 송신에 대한 정보 전달  \* Step 10: $TRANSREQ 전문에서 파일이름 정보는 특수문자 변환, Utf8로 변환과정 거침 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일보내기(FT) | 취소(CAN) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FT.CAN** | **기능번호** | **FT.FT.5** |
| **기능설명** | 송신동작을 진행중에 중지하고 hsz 임시파일을 제거 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: “TRANSREQ” 이름의 사용자 관리 이벤트를 송신동작 시작때 등록  \* Step 2: 해당 이벤트가 있는지 hsz 파일 생성, Server로 전송하는 과정에 주기적으로 체크  \* Step 3: 파일 송신중 ProgressBar UI에서 “취소”를 누르면, “TRANSREQ” 이름의 사용자 관리 이벤트 제거  \* Step 4: hsz 파일 생성, Server로 전송하는 동작이 중지됨  \* Step 5: hsz 파일이 생성되어 있다면 파일삭제  \* Step 6: 송신 동작이 중지 되었음을 UI상에 표현 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일추가(FA) | window탐색기 Drag&Drop(WDD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FA.WDD** | **기능번호** | **FT.FA.1** |
| **기능설명** | 윈도우 탐색기에서 파일을 파일전송 List 로 추가 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: 파일전송 List로 WM\_DROPFILES message가 전달  \* Step 2: 파일전송 List Control의 message Handler 에서 전달된 파일들 경로 목록을 저장  \* Step 3: 상세 처리부 호출, 다음과 같은 동작들 처리  \* Step 4: Folder일경우, 하위 Folder에 있는 모든 파일들을 찾아서 목록에 저장  \* Step 5: 파일 검사 동작으로 연결됨 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일추가(FA) | 파일등록 버튼사용(FBU) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FA.FBU** | **기능번호** | **FT.FA.3** |
| **기능설명** | 파일등록 버튼으로 뜨는 FileDialog 상자에서 파일 추가 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: FileDialog 상자를 띄워, 파일들을 선택할 수 있게 한다. (Folder 추가불가)  \* Step 2: 전달 받은 파일들 경로 목록 저장  \* Step 3: 파일 검사 동작으로 연결됨 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일추가(FA) | window탐색기 마우스 우클릭 (WER) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FA.WER** | **기능번호** | **FT.FA.4** |
| **기능설명** | 탐색기에서 우클릭을 통해 폴더, 파일들을 한꺼번에 추가 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: OS 탐색기에 등록된 DLL에서 우클릭때, 선택된 파일들 목록을 권한이슈가 없는 Folder에 “연월일시분초.txt” 파일로 저장  \* Step 2: txt파일을 Hidden으로 설정  \* Step 3: NetLink 에 우클릭때 선택된 파일개수, txt파일경로 정보를 전달  \* Step 4: NetLink 에서 해당 파일을 열어 목록을 읽어드림  \* Step 5: 목록들 중에 Folder가 있는지 확인해서 있으면 하위에 있는 모든 파일들을 찾아서 목록에 추가  \* Step 6: 파일 검사 동작으로 연결됨 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일추가(FA) | window탐색기 마우스 우클릭 자동 전송(WERA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FA.WERA** | **기능번호** | **FT.FA.5** |
| **기능설명** | 마우스 우클릭 파일 첨부 후 자동전송 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: OS 탐색기에 등록된 DLL에서 우클릭때, 선택된 파일들 목록을 권한이슈가 없는 Folder에 “연월일시분초.txt” 파일로 저장  \* Step 2: txt파일을 Hidden으로 설정  \* Step 3: NetLink 에 우클릭때 선택된 파일개수, txt파일경로 정보를 전달  \* Step 4: NetLink 에서 해당 파일을 열어 목록을 읽어드림  \* Step 5: 목록들 중에 Folder가 있는지 확인해서 있으면 하위에 있는 모든 파일들을 찾아서 목록에 추가  \* Step 6: 우클릭 파일 첨부 후 자동전송 설정 상태인지 확인  \* Step 7: 파일검사 동작으로 연동  \* Step 8: 파일송신 동작으로 연동 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일추가(FA) | Long File Name (LFN) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FA.LFN** | **기능번호** | **FT.FA.6** |
| **기능설명** | 사용하는 OS system(MAC/Window/Linux/…기타)에서 지원하는 최대 파일이름 길이 허용 | | |
| **입력** |  | | |
| **출력** |  | | |
| OS 마다 탐색기, Finder 등에서 지원하는 파일이름, 전체경로 길이가 다르므로 수신 받는쪽을 고려할 때 길이가 가장 짧은 쪽을 기준으로 파일명 / 전체경로 길이 제한을 해야 함.  단일 OS간에 송수신이라면 해당 OS에서 지원하는 최대 파일명 / 전체경로를 지원  파일이름을 저장하는 모든 stack 배열 buffer를 크게 사용(HszFrame64A 같은 Type 사용)  긴파일명 사용시 수신동작 과정  \* Step 1: 파일 수신할 내용이 있다는 $PUSHFILEINFO 통신전문이 Server에서 옴  \* Step 2: OS에서 지원하는 최대 파일이름을 지원하는지 확인  \* Step 3: 수신받는 위치를 사용자가 정하도록 UI단에 $PUSHFILEPATH 라는 ommand 전달  \* Step 4: 이번 전송에 TRANSSEQ 가 있는지 확인, 없으면 동작 중지  \* Step 5: 이전 TRANSSEQ와 같은지 확인, 같으면 동작 중지  \* Step 6: 이번 수신때에 전송하려는 파일경로의 길이와 정보를 얻어와 계산하는데 사용  \* Step 7: OS에서 지원하는 수치보다 높으면 수신하는 다른 경로를 선택하게 UI에 표현  \* Step 8: 취소선택시 전송취소($SENDCANCEL) command를 서버에 전달  \* Step 9: 적절한 경로선택시, 해당 경로 buffer에 복사해둠  \* Step 10: $AUTODOWNLOAD 전문 Server로 송신, Server가 $AUTODOWNLOAD\_ACK과 함께 파일을 보냄  \* Step 11: “위에서 선택한 경로 + temporary” 에 hsz 파일 다운로드  \* Step 12: $AUTODOWNLOAD, Server로 전달 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일추가(FA) | 파일 / Folder(FNF) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FA.FNF** | **기능번호** | **FT.FA.7** |
| **기능설명** | Folder와 그 내부에 있는 파일들도 한꺼번에 추가 | | |
| **입력** |  | | |
| **출력** |  | | |
| 파일전송때, 파일들은 파일에 대한 정보 + 파일 Data가 같이 전송  파일전송때, 폴더들은 폴더에 대한 정보만 전송  \* Step 1: 추가된 항목이 파일인지 Folder인지 확인  \* Step 2: Folder 이라면 현재 Folder에 있는 모든 파일들 탐색해서 현재 경로와 함께 List에추가  \* Step 3: 현재 Folder안에 Folder가 있다면 해당 Folder 안에 모든 파일들도 탐색해서 List에 추가 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일포워드(FF) | 파일포워드(FF) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FF.FF** | **기능번호** | **FT.FF.1** |
| **기능설명** | 자신뿐 아니라 대상자로 추가된 사람들에게도 송신한 파일들이 (수동)수신됨 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: 파일포워드로 파일을 다운로드 받을 사람들 파일전송UI에서 추가  \* Step 2: 파일추가 및 송신  \* Step 3: $TRANSREQ 전문 송신때, FORWARDUSERID 항목에 받을 사람들 userID를 추가 전송  \* Step 4: 전송 / 결재관리에 “파일 전달” 컬럼추가로 나타남, 전송관리에는 “다운로드”, “전송취소” 버튼이 추가로 나타남  \* Step 5: 전송 / 결재관리 > 상세조회에서 수신자가 1명이 아니라 여러명 LIST 형태로 표현됨  \* Step 6: 파일을 송신한 사람이 수신을 먼저 해야만 포워드 대상자들도 해당 파일들을 수신가능함  \* Step 7: 파일을 송신한 사람이 $MANUALDOWNLOAD 전문을 보내, $MANUALDOWNLOAD\_ACK 로 해당 파일을 다운로드 받게 되면 포워드 대상자들도 해당 파일을 동일한 전문을 보내 수신 받을 수 있다  \* Step 8: 포워드 대상자들이 “전송관리”, “전송 상세보기” 에서 “수신취소”를 선택하면, $FORWARD\_CANCEL 전문을 서버에 보내어 포워드 받기를 거부할 수 있다.  ※ “파일만기일”, “다운로드 가능횟수“ 정보는 여러 번 다운로드 기능으로 파일포워딩 기능과는 무관. 포워딩 받는 사람들이 수신을 다 받을 때까지 해당 파일은 서버에서 지워지지 않음 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 보안이미지(SI) | CAPTCHA(CHA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.SI.CHA** | **기능번호** | **FT.SI.1** |
| **기능설명** | 파일송신때, 서버에서 CAPTCHA 이미지를 받아와 바르게 입력하면 파일을 보낸다 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: 파일보내기로 파일들을 보내려고 할 때, $SECUREIMAGE 전문을 서버로 송신  \* Step 2: $SECUREIMAGE\_ACK를 서버에서 수신받을 때, 보안이미지 Data와 그와일치하는 string(알파벳대문자+숫자) 값을 받음  \* Step 3: 보안이미지는 Data를 BASE64로 Decoding 한 후에 띄워줌  \* Step 4: UI 대화상자를 띄워 받은 보안이미지를 보여주고 string을 입력받음  \* Step 5: 입력받은 string은 알파벳 대문자화해서 $SECUREIMAGE\_ACK로 받은 string과 일치하는지 확인  \* Step 6: 바르게 입력되었다면, 파일보내기 과정 계속진행, 실패하면 파일보내기 중지  \* Step 7: 만약 이미지가 정확하게 보이지 않다면, Refresh 로 $SECUREIMAGE 를 다시 보내어 갱신할 수 있음 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 사후결재(AA) | 사후결재(AA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.AA.AA** | **기능번호** | **FT.AA.1** |
| **기능설명** | 자료전송으로 파일을 먼저 보내고 이후에 결재를 받는 기능. server에서 지정해놓은 조건에 해당해야 사용자가 사후결재를 사용할 수 있는 checkbox가 나타남. | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: Server에서 설정해놓은 HOLIDAY, AFTERAPPROVE 값을 보고 사후결재를 사용하게 해줄지 결정  \* Step 2: agent / PC를 그냥켜두고 있는 사람들이 있으므로 매시간 0분 1초마다 HOLIDAY, AFTERAPPROVE 값으로 사후결재를 사용할 수 있도록 할지 판단  \* Step 3: 사후결재를 사용하도록 체크박스를 선택하고 파일들을 송신하면, 반대망으로 파일들이 송신.  \* Step 4: 결재는 이후에 결재자에게 받아야 하며, 결재자가 사후결재로 이미 전송이된 파일들에 대해 미리보기를 통해 파일을 확인할 수 있는 날짜는 서버의 AFTERAPPROVEFILE 설정에 따름 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일명 치환(FNC) | 파일명 특수문자 치환(FNC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FNC.FNC** | **기능번호** | **FT.FNC.1** |
| **기능설명** | 파일명에서 서버 DB 쿼리에 사용이 불가한 문자를 치환 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: 파일명에서 서버 DB 쿼리에 사용이 불가한 문자를 치환해서 전송  \* Step 2: 다음 특수문자들 적용, 정확한건 소스에서 파악  ! :^EM^ , @ :^AT^ , # :^SH^ , $ :^DL^ , % :^PC^ , & :^AP^ , \* :^AS^ , (:^LR^ , ) :^RR^ , ` :^TD^ , - :^DS^ , + :^PL^ , = :^EQ^ , ; :^SC^ , ' :^SQ^  \* Step 3: 수신후 치환된 문자들 원래대로 치환해서 UI 및 data로 사용  Ex.1) hsz 압축을 진행할때, HszFrame64A 의 fname 에 파일이름을 복사할 때, 치환해서 Utf8로 변환해서 저장 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일변환(FC) | Office To Pdf(OP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FC.OP** | **기능번호** | **FT.FC.1** |
| **기능설명** | office 파일들을 pdf로 변환해서 송신, Office 가 PC에 설치되어 있을때에만 동작 | | |
| **입력** |  | | |
| **출력** |  | | |
| Microsoft Office2010 이상의 Office가 PC에 설치되어 있어야만 변환기능 가능  서버에 PDFEXCHG 값을 설정하므로 동작관련 설정가능   * ex) 1/pdf/2/20/0   Office 파일 > pdf 로 변환 후 전송, 20MB이하만 변환전송, 초과되는 파일은 전송안됨.  \* Step 1: 이 기능을 사용하도록 PDFEXCHG 값이 설정되어 있는데, MS Office가 설치되어 있지 않다면 NetLink 실행시 “[Microsoft Office]가 설치 되어 있지 않아, Office파일을 pdf파일로 변환 전송할 수 없습니다.” 라는 메시지가 나타난다.  \* Step 2: 이 상태에서 Office 파일을 전송하려고 하면 같은 메시지가 나타난다.  \* Step 3: 변환은 “파일보내기” 버튼을 눌렀을 때, 시작되며, 변환과정 또한 UI progressbar로 표현됨  \* Step 4: 변환된 pdf 파일은 현재 NetLink가 설치된 곳의 PdfTemp Folder에 생성  \* Step 5: MS Office에서 제공하는 COM Type DLL의 api를 사용하여 변환동작을 수행  \* Step 6: 각 파일에 맞는 Program들이 떠서 Office 파일들을 PDF 파일로 변환  \* Step 7: a.pptx 파일은 a.pptx.pdf 라는 이름의 pdf 파일로 전송  \* Step 8: password가 걸려있거나 설치되지 않은 상용 Font를 사용하는 Office 파일은 변환실패 현상있음 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일변환(FC) | Office To Tiff(OT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FC.OT** | **기능번호** | **FT.FC.2** |
| **기능설명** | office 파일들을 pdf로 변환, Tiff로 다시 변환해서 송신, Office 가 PC에 설치되어 있을때에만 동작 | | |
| **입력** |  | | |
| **출력** |  | | |
| Microsoft Office2010 이상의 Office가 PC에 설치되어 있어야만 변환기능 가능  서버에 PDFEXCHG 값을 설정하므로 동작관련 설정가능   * ex) 1/tiff/2/20/0   Office 파일 > pdf > tiff 로 변환 후 전송, 20MB이하만 변환전송, 초과되는 파일은 전송안됨.  3th-party lib : MuPDFLib-x86.dll, libtiff.dll (경로명중에 한글이 있으면 변환 error나는 Bug 있음.)  \* Step 1: Office 파일을 PDF 파일로 변환하는 과정은 위의 기능코드 **FT.FC.OP** 의 과정과 동일  \* Step 2: pdfTemp Folder에 있는 변환된 \*\*\*.pptx.pdf 파일을 A.pdf 파일로 이름변경  \* Step 3: A.pdf에서 A.tiff로 변환 파일을 생성(300dpi 사용)  \* Step 4: A.tiff 를 \*\*\*.pptx.tiff 로 파일이름 변경 / 파일전송  \* Step 5: pdfTemp Folder에 있는 모든 파일들 삭제 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 파일변환(FC) | Office To Tiff(EPH) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FC.EPH** | **기능번호** | **FT.FC.3** |
| **기능설명** | eml To Pdf, eml To html(EPH) | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 기타(ETC) | 특정 수신위치 설정(RPS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.ETC.RPS** | **기능번호** | **FT.ETC.1** |
| **기능설명** | 파일 전송 시 보안웹하드로 수신 여부 설정 | | |
| **입력** |  | | |
| **출력** |  | | |
| 참고 site : HMC  NetLink에서 파일을 송신하면 서버를 거처 NetLink에게 오지 않고, 고객사 웹하드로 파일이 전달  \* Step 1: 파일 전송때, “웹하드전송” 이라는 checkbox를 선택, 파일보내기 실행  \* Step 2: $TRANSREQ 를 보낼 때, RECVPOS 값을 “1” 로 설정해서 서버로 파일과 전문 전송  \* Step 3: 해당 파일은 압축된 hsz가 아니라 원본 파일형태로 웹하드로 전달됨  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 기타(ETC) | 즐겨찾기(FAV) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.ETC.FAV** | **기능번호** | **FT.ETC.2** |
| **기능설명** | 즐겨찾기 전송 | | |
| **입력** |  | | |
| **출력** |  | | |
| 메뉴 > 도구 > 즐겨찾기 전송 :  PC에 로그인한 계정의 즐겨찾기 Folder에 있는 \*.url 파일들을 반대망으로 보냄  적용 site : 화학연구원  \* Step 1: PC에 로그인한 계정의 즐겨찾기 Folder에 있는 \*.url 파일들 목록을 저장  \* Step 2: NetLink가 설치된 곳 하위 Folder Work(C:\HANSSAK\SecureGate\Work\SGFavorites.pcf) 에 \*.url 파일들을 저장하는 SGFavorites.pcf 파일을 생성  \* Step 3: SGFavorites.pcf 파일은 Header 정보인 버전정보, \*.url 파일 갯수를 먼저 저장  \* Step 4: SGFavorites.pcf 파일에 \*.url 파일에 대한 데이터타입, 암호화한 파일이름, 파일크기, 파일내용을 반복적으로 저장  \* Step 5: 파일 전송때 사용한 $TRANSREG 전문으로 SGFavorites.pcf 파일을 송신  \* Step 6: 결재정보를 빼고 넣으므로 결재없이 전달, 제목/설명에 “즐겨찾기 전송” 이라고 명시  \* Step 7: 수신된 SGFavorites.pcf 파일을 해제, PC에 로그인한 계정의 즐겨찾기 Folder에 url 파일로 저장 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 기타(ETC) | 전송 제한(FAD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.ETC.FAD** | **기능번호** | **FT.ETC.3** |
| **기능설명** | 문서파일(한글오피스), zip 파일 암호설정된 파일 전송 제한 | | |
| **입력** |  | | |
| **출력** |  | | |
| zip파일 - 참고 site : 농협생명 (3th-party lib : zlib)  \* Step 1: 파일 검사 > 파일 유효성 확인후 zip 내부 검사때, 암호걸린 ZIP 파일인지 확인 / 차단  문서파일(한글오피스) - 참고 site : 심평원  \* Step 1: 파일 검사 > 파일 유효성 확인때, 한글오피스파일(hwp, hwpx, hwt)이면 1MB Data를 읽어드림  \* Step 2: Hwp의 Header 중 중요Data("\x48\x57\x50\x20\x44\x6F\x63\x75\x6D\x65\x6E\x74\x20\x46\x69\x6C\x65") 를 찾아냄  \* Step 3: Hwp Header 중요 Data 시작지점에서 36byte 뒤 값이 0x02 와 AND 연산시 값이 있으면 암호가 걸린 것으로 판단  \* Step 4: 파일 첨부 실패 처리 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 기타(ETC) | 전송 차단(TD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.ETC.TD** | **기능번호** | **FT.ETC.4** |
| **기능설명** | 특정 부서에서 승인대기 상태인 사후결재건이 MAX 값을 초과 하면 전송 차단 | | |
| **입력** |  | | |
| **출력** |  | | |
| 참고 site :  \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 기타(ETC) | 결재 예외 전송(AET) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.ETC.AET** | **기능번호** | **FT.ETC.5** |
| **기능설명** | 특정 확장자의 파일에 대해서만 결재 예외 전송 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 기타(ETC) | 원본파일 삭제(OFR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FT.HSZ** | **기능번호** | **FT.FT.6** |
| **기능설명** | 특정 확장자의 경우 원본파일 삭제 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 기타(ETC) | FTP 파일 업로드(FFU) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.FT.FFU** | **기능번호** | **FT.FT.7** |
| **기능설명** | 파일을 FTP 서버로 전송 | | |
| **입력** |  | | |
| **출력** |  | | |
| 참고 site : SK실트론, EAI 결재연동에 사용  \* Step 1: 송신파일을 망연계 서버에 전송후, 결재자가 있는 상태이면 결재연동 FTP에 접속  \* Step 2: ftp 서버의 \VDI\년\월\일 형태 경로에 원본 파일을 upload | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일송신(FT) | 기타(ETC) | zip 파일 비번 입력(ZFPI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FT.ETC.ZFPI** | **기능번호** | **FT.ETC.8** |
| **기능설명** | 파일 수신 시 입력받은 비번으로 zip 암호화 압축을 하기 위해 사용자로부터 비번을 입력받음 | | |
| **입력** |  | | |
| **출력** |  | | |
| 참고 site : 한국투자저축은행  \* Step 1: 파일 송신때, 사용자로부터 zip 파일의 password를 입력받음  \* Step 2: password를 $TRANSREQ 에 추가해서 송신  \* Step 3: $PUSHFILEINFO 를 받으면, TransSeq 값을 이용해 TBL\_TRANSFER\_REQ\_INFO, TBL\_TRANSFER\_REQ\_HIS 쪽에 Query를 던져서 ZipPassword를 받아옴  \* Step 4: 파일을 수신받고, hsz에서 해제한 후 파일 1개마다 보안zip압축 프로그램으로 압축을 하는데, 앞에서 받은 ZipPassword를 사용함 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 공통(COM) | 복호화(DEC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.COM.DEC** | **기능번호** | **FR.COM.1** |
| **기능설명** | 수신파일 AES256 복호화 | | |
| **입력** |  | | |
| **출력** |  | | |
| Hsz 파일 생성과정 참조  \* Step 1: hsz 파일을 해제해서 송신한 파일 생성 시작  \* Step 2: HszHead 를 읽어 hsz 파일에 대한 정보를 읽음  \* Step 3: HszFrame64A Data 읽어서 AES256 복호화 적용  \* Step 4: HszFrameSlot Data 읽어서 length에 만큼의 Data를 읽어 AES256 복호화 적용  \* Step 5: AES256 복호화에 사용하는 key는 $AUTODOWNLOAD\_ACK에서 받은 FileKey 사용 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 공통(COM) | 무압축해제/무복호화(NZE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.COM.NZE** | **기능번호** | **FR.COM.2** |
| **기능설명** | hsz파일에서 HszHead의 hszOption을 보고 압축해제 및 복호화 단계를 생략 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: 수신받은 hsz 파일에서 HszHead의 hszOption 값 읽어냄  \* Step 2: hszOption 값을 2진수화해서 아래과정으 거침으로 hsz 파일이 어떻게 저장되어 있는지를 알게됨  \* Step 3: 2진수 값에서 제일 오른쪽 값, UTF8(1) / EUCKR(0)  \* Step 4: 2진수 값에서 오른쪽 2째값, 32bit(0) / 64bit(1)  \* Step 5: 2진수 값에서 오른쪽 3째값, 압축(0) / 무압축(1), hsz 파일을 Z압축해제를 사용할지 말지 결정  \* Step 6: 2진수 값에서 오른쪽 4째값, 암호화(0) / 무암호화(1), HszFrame64A와 HszFrameSlot뒤에 붙은 Data를 AES256 복호화할지 결정  \* Step 7: $AUTODOWNLOAD\_ACK 로 받은 FileKey를 HsAes256DecryptKey로 얻어낸 값으로 AES256 복호화를 진행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 공통(COM) | 수신과정(RP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.COM.RP** | **기능번호** | **FR.COM.3** |
| **기능설명** | 파일 수신할 때, 전체 진행 상황 및 개별 진행 정보 표현 | | |
| **입력** |  | | |
| **출력** |  | | |
| 파일이 서버에서 수신되는 과정에 전체 진행 상황 및 개별 진행 정보 표현에 대해 설명  \* Step 1: $PUSHFILEINFO 가 서버에서 옴  \* Step 2: NetLink로 지속적으로 내려오므로 TransSeq 값을 보고 중복되면 특별한 처리를 진행하지 않음  \* Step 3: $AUTODOWNLOAD 를 서버에 전달  \* Step 4: 서버에서 $AUTODOWNLOAD\_ACK 으로 hsz 파일과 정보를 보내줌  \* Step 5: ClientMem내 DownFileInfo64의 totalpart 값이 0 일때, 수신이 시작 됐다고, UI쪽에 hsz 파일크기인 totalsize 정보와 함께 정보를전달  \* Step 6: $AUTODOWNLOAD\_ACK 가 올 때 마다 ClientMem내 DownFileInfo64의 cursize 정보를 계속 합해서 UI 단에 전달함으로서 전체크기 대비 현재 어디까지 수신이 됐는지 UI상으로 표현이 가능함  \* Step 7: 수신받은 hsz 파일을 압축해제 시작  \* Step 8: 압축해제시작 정보 전달 때, 원본 파일 Size 전체크기 전달  \* Step 9: 압축해제 진행과정 정보 UI에 전달, 압축해제진행된 Data 크기 전달  \* Step 10: 압축해제 완료되면 해당 메시지 UI에 전달 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 공통(COM) | 수신위치 및 용량 확인(DS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.COM.DS** | **기능번호** | **FR.COM.4** |
| **기능설명** | PC 용량부족 시, 다운로드 일시중지 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: $AUTODOWNLOAD\_ACK 로 파일Data와 정보를 받기 시작할 때, hsz 파일의 크기인 TotalSize 정보가 같이 옴  \* Step 2: hsz 파일을 Down 받는 경로정보에서 Drive 경로를 얻어 해당 Drive에 사용가능한 용량을 찾음  \* Step 3: TotalSize \* 2.5 보다 사용가능한 용량이 크지 않으면, 사용자에게 관련 message를 전달하고 NetLink는 종료됨  \* Step 4: 압축해제를 진행하는 과정에서 각 개별파일의 크기\*2.5 보다 Drive에 사용가능한 용량이 작으면, 사용자에게 관련 message를 전달하고 NetLink는 종료됨 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 공통(COM) | 동일이름 파일 변경(FNC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.COM.FNC** | **기능번호** | **FR.COM.5** |
| **기능설명** | 같은 경로에 이미 같은 이름의 파일이 있을 경우, "다운받는 파일의 이름 + (숫자).확장자"로 변경 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: hsz 파일을 압축해제 진행  \* Step 2: 파일을 다운로드 하는 위치에 HszFrame64A의 fname에 있는 파일 혹은 폴더와 동일한 것이 이미 있는지 확인  \* Step 3: 없다면 해당항목 그냥 압축해제 진행  \* Step 4: 있으면, 압축해제해서 생성할 폴더 및 파일이름을 “이름” 에서 “이름 (1)” 형식으로 바꿈  \* Step 5: 만약, “이름 (1)”가 이미 있다면 “이름 (2)” 과 같은 식으로 숫자를 증가해서 파일이름을 만듬  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 공통(COM) | 다운로드 파일 관리(DFM) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.COM.DFM** | **기능번호** | **FR.COM.6** |
| **기능설명** | 삭제주기에 따른 다운로드 파일 삭제 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: Server 설정값, PCDELETECYCLE 에 따라 수신받은 파일들을 삭제하는 주기가 달라짐  \* Step 2: “0” 일때는 동작하지 않음. 값은 시간값 기준임  \* Step 3: 삭제동작을 할 시간인지 체크하는 동작, 30초 넘어서 한번씩 실행함  \* Step 4: NetLink에 설정된 수신경로에서 파일들을 찾음  \* Step 5: 파일생성시간을 확인해서 PCDELETECYCLE 값이 1일때, 지금시간보다 1시간이전에 생성된 파일이면 파일을 삭제함  \* Step 6: NetLink로 다운로드한 파일을 몇시간 지나면 삭제할지를 설정하게 됨  Cf) 중요한 파일이 있느 곳에 파일 다운로드 경로를 설정하게 되면 중요한 파일을 잃을 수 있음 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 공통(COM) | 유효성 확인(HC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.COM.HC** | **기능번호** | **FR.COM.7** |
| **기능설명** | 수신하는 파일이 송신할때의 파일과 동일한지 hsz file의 Hash를 check | | |
| **입력** |  | | |
| **출력** |  | | |
| Server에서 받은 FILECHECKSUM 값이 1:MD5, 2:SHA384, 3:SHA512 로 Hash check  \* Step 1: hsz 파일 수신완료  \* Step 2: FILECHECKSUM 값에 맞는 Hash값 얻음  \* Step 3: Server에서 보내준 $AUTODOWNLOAD\_ACK 에 있는 FileMd5 값과 비교  \* Step 4: 같으면 정상적인 hsz 파일 압축해제 진행  \* Step 5: 다르면, error message 출력, 받은 hsz 파일 삭제 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 수동다운로드(MD) | 수동다운로드(MD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.MD.MD** | **기능번호** | **FR.MD.1** |
| **기능설명** | 파일을 수신 받는 시점을 사용자가 원할때에 버튼을 눌러서 결정 | | |
| **입력** |  | | |
| **출력** |  | | |
| Server에 MANUALDOWNLOAD 값이 0이 아니면 수동다운로드 사용으로 설정  \* Step 1: 파일을 송신후 반대망 서버로 파일이 도착후, “전송관리”에서 “다운로드”를 누르면 파일이 수신됨  \* Step 2: 다운로드를 눌렀을 때, $MANUALDOWNLOAD 를 서버로 보냄  \* Step 3: $MANUALDOWNLOAD\_ACK 로 Server에서 파일과 정보를 보냄  \* Step 4: hsz 파일 생성 및 압축해제후, $MANUALDOWNEND 를 Server로 보냄  \* Step 5: $MANUALDOWNEND\_ACK 를 Server 에서 받음 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 수동다운로드(MD) | 수신시 경로 변경(MDC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.MD.MDC** | **기능번호** | **FR.MD.2** |
| **기능설명** | 파일을 수신 받을 때마다 받고 싶은 경로를 지정 | | |
| **입력** |  | | |
| **출력** |  | | |
| 수동 다운로드로 파일을 받을 때 마다, 파일을 저장할 곳을 사용자가 지정할 수 있도록 UI가 뜨고, 지정하면 수동 다운로드가 시작된다  \* Step 1: 전송관리에서 수신 받기 원하는 건수를 선택, 다운로드 선택  \* Step 2: SELECT O\_TRANS\_SEQ, O\_TITLE, O\_LEN, O\_FSIZE FROM FUNC\_NL\_MAXFILELENGTH(transseq) 를 사용하여 파일이름 중 최대로 긴 걸 찾아서 OS에서 지원하는 경로, 파일길이가 넘지 안는지 확인  \* Step 3: 저장용량이 충분한지 확인  \* Step 4: 수동 다운로드 시작 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 수동다운로드(MD) | 멀티다운로드(MDW) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.MD.MDW** | **기능번호** | **FR.MD.3** |
| **기능설명** | 최대 다운로드 가능 횟수 안에서 몇번이고 다시 다운가능, 파일 만기일, 다운로드 가능횟수 넘길 수 없음. | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: $Bind\_ACK 에서 받은 DOWNLIMITCOUNT 값으로 최대 다운 가능횟수 파악  \* Step 2: TransSeq 값과 함께 $DOWNLOAD\_COUNT 를 Server 로 보내어 $DOWNLOAD\_COUNT\_ACK를 받아와 다운로드한 횟수 파악  \* Step 3: 선택한 전송건에 대해 현재 다운로드 가능 횟수는 { 최대 다운가능횟수 - 다운로드한 횟수 } 임  \* Step 4: $FILE\_TRANSLIST\_ACK 로 받은 만기일 날짜와 현재날짜를 비교해서 다운로드 가능유무 파악  \* Step 5: 만기일 날짜가 없는 경우에는 요청일+시스템보관일과 현재날짜 비교해서 가능유무 파악 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 수동다운로드(MD) | 파일만기일 설정(FEC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.MD.FEC** | **기능번호** | **FR.MD.4** |
| **기능설명** | 멀티 다운로드 수행 시 사용자가 원하는 기간 만큼 변경 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일수신(FR) | 수동다운로드(MD) | 파일포워딩 수신자 다운로드(FFRD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FR.MD.FFRD** | **기능번호** | **FR.MD.5** |
| **기능설명** | 파일포워딩 사용 시 수신자의 경우 버튼을 눌러 파일을 수신받음 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | 위변조 검사(FA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.FA** | **기능번호** | **FC.FC.1** |
| **기능설명** | 전송하려는 파일의 확장자가 원래의 것이 아닌지 확인 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | 확장자 black/white List 검사(ELA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.ELA** | **기능번호** | **FC.FC.ELA** |
| **기능설명** | 전송하려는 파일의 확장자가 허락된 것인지 아닌지 확인 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | 숨김 파일인지 검사(HFA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.HFA** | **기능번호** | **FC.FC.3** |
| **기능설명** | 숨김파일로 설정된 파일이면 전송하고 싶은 목록에 추가 불가 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | 암호걸린Zip파일인지 검사(PZA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.PZA** | **기능번호** | **FC.FC.4** |
| **기능설명** | site 요구에 따라 zip에 암호가 걸린것만 전송 혹은 아닌 것만 전송하도록 검사 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | zip내부 확장자 제한 검사(IZXA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.IZXA** | **기능번호** | **FC.FC.5** |
| **기능설명** | zip 내부 확장자들을 List Up해서, 확장자 black/white List 검사 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | zip해제해서 파일검사(UZFA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.UZFA** | **기능번호** | **FC.FC.6** |
| **기능설명** | disk용량, DRM여부, 확장자, 위변조, zip내부zip, 바이러스 검사함 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | DRM 연동때 권한, 대상 검사(DFA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.DFA** | **기능번호** | **FC.FC.7** |
| **기능설명** | drm 해재 권한, 대상 확인해서 정책따라 drm 적용 / 해제, 서버연동 자주있음 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | DLP 연동때 권한, 대상 검사(LPFA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.LPFA** | **기능번호** | **FC.FC.8** |
| **기능설명** | dlp로 파일 확인, site 요구 정책 따라 파일 첨부 error 처리, 서버연동 자주있음 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | Anti-Virus 연동때, 검사(AVFA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.AVFA** | **기능번호** | **FC.FC.9** |
| **기능설명** | anti-virus Engine의 결과에 따라 파일 첨부 error 처리, 서버연동 자주있음 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | 파일전송 크기 검사(FSA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.FSA** | **기능번호** | **FC.FC.10** |
| **기능설명** | 개별 파일별로 설정된 크기를 넘는 파일은 전송불가 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | 파일전송 개수 검사(FCA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.FCA** | **기능번호** | **FC.FC.11** |
| **기능설명** | 첨부파일 개수가 최대치를 넘을때 파일 첨부 / 전송불가 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | 이미 등록된 파일인지 검사(FARA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.FARA** | **기능번호** | **FC.FC.12** |
| **기능설명** | 전체경로기준으로 중복 확인 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | 별도Log에 저장(ULGS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.ULGS** | **기능번호** | **FC.FC.13** |
| **기능설명** | 유효하지 않은 파일정보 / 사유정보 별도 log파일에 저장, 솔루션운영자가 참조 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1:  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10: | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | 파일추가 실패사유 DB 저장(FAFDB) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.FAFDB** | **기능번호** | **FC.FC.14** |
| **기능설명** | 파일 첨부 실패사유 서버 DB에 기록 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: 송신위해 추가되는 파일  \* Step 2:  \* Step 3:  \* Step 4:  \* Step 5:  \* Step 6:  \* Step 7:  \* Step 8:  \* Step 9:  \* Step 10:  eFAREG=1, // 등록된 파일  eFADRM, // DRM  eFADLP, // DLP  eFAEXT, // 확장자제한  eFAZIP, // zip파일내 확장자  eFACHG, // 파일변경  eFAVIRUS, // 바이러스검출  eFAFileSize, // 파일사이즈  eFANotFound, // 파일 찾기 실패  eFAHidden, // 숨김파일  eFAZipPW, // zip 파일 비번 있을 때  eFAZipNotPW, // zip 파일 비번 없을 때  eFAZipError, // zip 파일 손상 또는 zip 파일이 아닌경우  eFAEMPTY, // 빈파일  eFAUNKNOWN, // 알수없는파일형식  eFAEML, // EML파일과 다른파일을 함께 등록할 경우  eFAEMPTY\_ATTACH, // 빈파일(첨부파일)  eFAUNKNOWN\_ATTACH, // 알수없는파일형식(첨부파일)  eFACHG\_ATTACH, // 파일변경(첨부파일)  eFAEXT\_ATTACH, // 확장자제한(첨부파일)  eFAZIP\_ATTACH, // zip파일내 확장자(첨부파일)  eFAEML\_ONLYONE, // EML 파일등록 2건이상일때  eFAEMLTOPDF\_ERROR, // EML to PDF 변환오류  eFAOfficeSizeError, // Office > pdf 변환하려는 파일이 설정되크기보다 클 경우  eFAOfficeNoinstalled, // Office 설치않되어있음. 파일변환기능사용. Office파일 전송하려할 경우  eFADLPERR // 개인정보 검출에러  eFAUnZipOutOfSpace //UnZip Disk용량부족  eFAUnZipLengthOver //UnZip Length Over  eFAUnZipCheckStop //UnZip 체크 중단  eFADAYCOUNTOVER = 51 // 일일 전송횟수 제한.  eFADAYSIZEOVER // 일일 전송사이즈 제한. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 파일검사(FC) | 파일 유효성 확인 (FC) | zip 파일 외 검사기능 (NZA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **FC.FC.NZA** | **기능번호** | **FC.FC.16** |
| **기능설명** | 7z, zipx, tar 전송파일 검사 추가 | | |
| **입력** |  | | |
| **출력** |  | | |
| 참고 site : 강원랜드  3th-party lib : 7z.dll  \* Step 1: 확장자가 7z, zipx, tar, zip 인 파일들은 7z.dll을 사용한 library로 읽음  \* Step 2: 내부 파일 개수 및 파일 목록 정보들을 읽음  \* Step 3: 파일 목록 정보 중에 FILEFILTER WhiteList에 없는 파일이 있거나, Black List에 있는 목록이 있으면 차단함  \* Step 4: 압축된 파일내부에 파일들 개수도 전송파일개수에 추가해서 송신파일 개수 제한에 적용 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | 파일포워드 전송관리(FFTM) | 파일전달 유무(FFIS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.FFTM.FFIS** | **기능번호** | **SD.FFTM.1** |
| **기능설명** | 전송관리에서 다른 사용자에게 파일 전달 유무가 조회된다. | | |
| **입력** | 조회 요청 | | |
| **출력** | 파일 전달 유무 조회 결과. | | |
| \* Step 1: 사용자가 다른 사용자에게 파일 전송을 하면 파일 전달 유무가 전송관리 화면에 조회된다.  \* Step 2: Agent 는 프로토콜 FILE\_TRANSLIST 를 사용하여 서버로 전송관리 조회를 요청한다.  \* Step 3: 서버로부터 수신받은 조회 응답 데이터를 파싱하여 SEQCOUNT 의 DATAFORWARDED  값을 확인한다.  \* Step 4: 전송관리 조회 화면에서 파일 전달 컬럼에 DATAFORWARDED 값에 따라 파일전달 유무가  표시된다.  Cf) DATAFORWARDED 0 – 파일전달 유무가 존재하지 않음. “-“ 로 표시.  1 – 파일포워딩한 사용자가 존재함. “전송” 으로 표시.  2 – 다른 사용자로부터 받는 수신자. “수신” 으로 표시. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | 파일포워드 전송관리 상세보기(FFD) | 수신자(RR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.FFD.RR** | **기능번호** | **SD.FFD.1** |
| **기능설명** | 전송관리 상세보기를 통해 수신자 리스트를 조회한다. | | |
| **입력** | 전송관리 상세보기 조회 요청. | | |
| **출력** | 수신자 리스트 | | |
| \* Step 1: 전송관리 조회 화면에서 파일포워딩 사용된 리스트를 선택한다.  \* Step 2: 항목이 선택된 상태에서 상세보기 버튼을 클릭하면 서버로 해당 건에 대해  TRANS\_DETAIL 프로토콜을 사용하여 상세보기 데이터 조회를 요청한다.  \* Step 3: 서버로부터 수신받은 데이터를 파싱하여 FORWARDUSERCOUNT 와 FORWARDUSERRECORD  값을 확인하다.  Cf) FORWARDUSERCOUNT – 수신자 개수.  FORWARDUSERRECORD – 수신자 리스트 정보  \* Step 4: 수신자 정보는 상세보기 창의 승인요청일 오른쪽에 표시되며 수신자가 2명 이상인 경우  수신자 리스트 화면이 출력되는 버튼이 활성화된다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | 파일포워드 결재관리(FFA) | 파일전달 유무(FFIS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.FFA.FFIS** | **기능번호** | **SD.FFA.1** |
| **기능설명** | 결재관리에서 승인요청자가 다른 사용자에게 파일 전달 유무가 조회된다. | | |
| **입력** | 조회 요청 | | |
| **출력** | 파일 전달 유무 조회 결과. | | |
| \* Step 1: 사용자가 다른 사용자에게 파일 전송을 하면 파일 전달 유무가 결재관리 화면에 조회된다.  \* Step 2: Agent 는 프로토콜 FILE\_APPROVE 를 사용하여 서버로 결재관리 조회를 요청한다.  \* Step 3: 서버로부터 수신받은 조회 응답 데이터를 파싱하여 SEQCOUNT 의 DATAFORWARDED  값을 확인한다.  \* Step 4: 결재관리 조회 화면에서 파일 전달 컬럼에 DATAFORWARDED 값에 따라 파일전달 유무가  표시된다.  Cf) DATAFORWARDED 0 – 파일전달 유무가 존재하지 않음. “-“ 로 표시.  1 – 파일포워딩한 사용. “사용” 으로 표시. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | 파일포워드 결재관리 상세보기(FFAD) | 수신자(RR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.FFAD.RR** | **기능번호** | **SD.FFAD.1** |
| **기능설명** | 결재관리 상세보기를 통해 승인요청자가 파일을 전달한 수신자 리스트를 조회한다. | | |
| **입력** | 전송관리 상세보기 조회 요청. | | |
| **출력** | 수신자 리스트 | | |
| \* Step 1: 결재관리 조회 화면에서 파일포워딩 사용된 리스트를 선택한다.  \* Step 2: 항목이 선택된 상태에서 상세보기 버튼을 클릭하면 서버로 해당 건에 대해  TRANS\_DETAIL 프로토콜을 사용하여 상세보기 데이터 조회를 요청한다.  \* Step 3: 서버로부터 수신받은 데이터를 파싱하여 FORWARDUSERCOUNT 와 FORWARDUSERRECORD  값을 확인하다.  Cf) FORWARDUSERCOUNT – 수신자 개수.  FORWARDUSERRECORD – 수신자 리스트 정보  \* Step 4: 수신자 정보는 상세보기 창의 파일위치 오른쪽에 표시되며 수신자가 2명 이상인 경우  수신자 리스트 화면이 출력되는 버튼이 활성화된다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL(PU) | URL 사용 신청가능 목록 검색(URL). |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PU.URL** | **기능번호** | **SD.PU.1** |
| **기능설명** | URL 사용 신청 화면을 통해 사용자가 원하는 URL 그룹명 및 URL 을 검색한다. | | |
| **입력** | 신청 가능 URL 그룹 및 URL 리스트 찾기 | | |
| **출력** | 검색하려는 URL 그룹 및 URL 아이템 이동. | | |
| \* Step 1: URL 사용신청화면에서 사용자가 검색하려는 URL 그룹명 및 URL 을 입력하고  찾기 버튼을 클릭하면 서버로부터 수신받았던 신청가능 URL 그룹 및 URL 리스트 정보에서  그룹 seq 또는 URL seq 정보를 탐색하여 왼쪽 트리에 검색된 해당 아이템으로  선택되어 표출된다.  \* Step 2: 사용자가 조회하려는 URL 그룹명 또는 URL 이 존재하지 않으면 사용자에게  검색 실패 안내 메시지를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL(PU) | URL 사용 현황 목록 검색(URU). |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PU.URU** | **기능번호** | **SD.PU.2** |
| **기능설명** | 허용 URL 현황 화면에서 사용자가 원하는 URL 그룹명 및 URL 을 검색한다. | | |
| **입력** | 승인된 URL 그룹 및 URL 리스트 찾기. | | |
| **출력** | 검색하려는 URL 그룹 및 URL 아이템 이동. | | |
| \* Step 1: 사용 가능한 URL 목록 화면에서 사용자가 검색하려는 URL 그룹명 및 URL 을 입력하고  찾기 버튼을 클릭하면 서버로부터 수신받았던 사용 승인된 URL 그룹 및 URL 리스트 정보에서  그룹 seq 또는 URL seq 정보를 탐색하여 검색된 해당 트리 아이템으로 선택되어 표출된다.  \* Step 2: 사용자가 조회하려는 URL 그룹명 또는 URL 이 존재하지 않으면 사용자에게  검색 실패 안내 메시지를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL 관리(PUM) | URL 조회(USD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUM.USD** | **기능번호** | **SD.PUM.1** |
| **기능설명** | URL 관리 조회 화면에서 URL 을 입력하여 사용 신청 URL 현황을 조회한다. | | |
| **입력** | URL 사용 신청 현황 조회 요청. | | |
| **출력** | URL 사용 신청 현황. | | |
| \* Step 1: URL 관리 조회 화면에서 사용자가 원하는 URL 을 입력하여 조회 버튼을 클릭한다.  \* Step 2: 조회 버튼 클릭 후 서버 DB 조회를 통해 대표 URL 명과 일치하는  URL 사용 신청 현황 조회 응답을 서버로부터 수신 받는다.  \* Step 3: 서버로부터 수신받은 응답 데이터를 파싱한 후 URL 관리 조회 화면에 조회 결과를  표출한다.  Cf) URL 사용 신청현황 DB 조회 쿼리   |  | | --- | | CREATE OR REPLACE FUNCTION public.func\_nl2\_urlreqsearch(  i\_userseq bigint,  i\_fromdate character varying,  i\_todate character varying,  i\_approvestatus character,  i\_url character varying,  i\_reqreason character varying,  i\_sql integer)  RETURNS TABLE(o\_transseq bigint, o\_approveseq bigint, o\_approvepossible character, o\_approvestatus2 character varying, o\_approvestatus character varying, o\_requsername character varying, o\_title character varying, o\_titleurl character varying, o\_reqdate character varying, o\_arpprovedate character varying)  LANGUAGE 'plpgsql'  COST 100  VOLATILE  ROWS 1000  AS $BODY$  DECLARE  V\_SQL varchar; -- 쿼리  WHAND VARCHAR; -- 검색조건이 있을때의 연결(WHERE OR AND)  BEGIN  V\_SQL:='  select a.trans\_seq  , cast ( 0 as bigint) req\_seq --a.req\_seq  , cast (0 as character) approvepossible  , cast (a.d\_pos as varchar) approvestauts2  , cast ((case when p.req\_flag =''2'' then ''11'' else p.approve\_flag end) as character varying) approvestauts  , cast (u.user\_name || (case when coalesce(u.user\_rank, u.user\_position, '''') !='' '' then ('' ''||coalesce(u.user\_rank, u.user\_position) ) else '''' end) as varchar)  , p.title  , p.content  , cast(p.request\_time as character varying)  , cast(p.approve\_time as character varying)  from  (  select \*  from tbl\_pcurl\_req\_info  union all  select \*  from tbl\_pcurl\_req\_his  where request\_time between '''||i\_fromdate||''' and '''||i\_todate||'''  ) p  , (  select trans\_seq, max(d\_pos) d\_pos, max(approve\_flag) approve\_flag  from (  select 1 d\_pos, \*  from tbl\_approve\_info  -- where approve\_user\_seq<> user\_seq  union all  select 0 d\_pos, \*  from tbl\_approve\_his  where appr\_req\_time between '''||i\_fromdate||''' and '''||i\_todate||'''  -- and approve\_user\_seq<> user\_seq  ) aa  group by trans\_seq  ) a  , tbl\_user\_info u  where p.trans\_seq = a.trans\_seq  and p.user\_seq=u.user\_seq  and u.user\_seq='||i\_userseq||'  ';    -- i\_approvestatus  -- 결재상태  IF i\_approvestatus IS NOT NULL AND i\_approvestatus!='' AND i\_approvestatus!='0' THEN  V\_SQL:=V\_SQL||CHR(13)||'AND (case when p.req\_flag =''2'' then ''11'' else p.approve\_flag end) = '''||i\_approvestatus||'''';    END IF;  -- 포함URL  -- character  IF i\_url IS NOT NULL AND i\_url!='' THEN  V\_SQL:=V\_SQL||CHR(13)||'AND func\_nl2\_urlcount(p.trans\_seq, '''||i\_url||''')>0';  END IF;  -- 사유  IF i\_reqreason IS NOT NULL AND i\_reqreason!='' THEN  V\_SQL:=V\_SQL||CHR(13)||'AND title like ''%'||i\_reqreason||'%''';  END IF;    V\_SQL:=V\_SQL||CHR(13)||'order by p.request\_time desc ';    IF i\_sql=1 THEN  RAISE NOTICE 'QUANTITY HERE IS %', V\_SQL; -- PRINTS 50  END IF;    RETURN QUERY EXECUTE  V\_SQL;  END;  $BODY$;  ALTER FUNCTION public.func\_nl2\_urlreqsearch(bigint, character varying, character varying, character, character varying, character varying, integer)  OWNER TO hsck; | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL 관리(PUM) | URL 신청 - 사용요청(UR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUM.UR** | **기능번호** | **SD.PUM.2** |
| **기능설명** | URL 신청 버튼을 클릭하여 신청가능한 URL 그룹 또는 URL리스트를 조회한다. | | |
| **입력** | 신청가능한 URL 그룹 또는 URL리스트를 조회 요청. | | |
| **출력** | 신청가능한 URL 그룹 또는 URL리스트 | | |
| \* Step 1: URL 관리 조회 화면에서 URL 신청 버튼 클릭 시 User Sequence 정보를 사용하여  서버 DB 조회를 통해 신청가능 URL 그룹 및 URL 리스트 정보를 얻어온다.  \* Step 2: 서버로부터 수신받은 신청가능 URL 그룹 및 URL 리스트 정보는 URL 사용 신청 화면  팝업 시 왼쪽 트리에 그룹별로 표출이 되고 사용자가 그룹 아이템을 클릭하면  그룹에 속한 URL 리스트 정보를 확인한다.  Cf) 신청가능 URL 그룹 및 URL 리스트 DB 조회 쿼리   |  | | --- | | CREATE OR REPLACE FUNCTION public.func\_nl2\_urluserassignemt(  i\_userseq bigint)  RETURNS TABLE(o\_grptype character, o\_pseq bigint, o\_urlseq bigint, o\_urlname character varying, o\_urldesc character varying, o\_lockstatus character)  LANGUAGE 'plpgsql'  COST 100  VOLATILE  ROWS 1000  AS $BODY$  DECLARE  V\_SQL varchar; -- 쿼리  WHAND VARCHAR; -- 검색조건이 있을때의 연결(WHERE OR AND)  BEGIN  V\_SQL:='  select distinct  cast(a.gtype as character),  cast(a.p\_seq as bigint),  cast(a.u\_seq as bigint),  cast(a.url as character varying),  cast(coalesce(a.urldesc, ''-'') as character varying),  cast(a.lockstatus as character)  from (  -- 허용된 URL 리스트  select 1 gtype, 0 p\_seq, g.urlgrp\_seq u\_seq, g.urlgrp\_name url, g.urlgrp\_desc urldesc, cast (1 as character) lockstatus, u.user\_seq  from tbl\_pcurl\_grp g  , tbl\_pcurl\_display\_map d  , tbl\_user\_info u  where g.urlgrp\_seq=d.urlgrp\_seq  and (case when g.target\_type=1 then u.user\_seq=d.target\_seq  when g.target\_type=2 then u.dept\_seq=d.target\_seq  when g.target\_type=0 then g.target\_type=g.target\_type  end  )  union all  select 0, g.urlgrp\_seq, p.url\_seq, p.url, p.url\_desc, cast (1 as character) lockstatus, u.user\_seq  from tbl\_pcurl\_grp g  , tbl\_pcurl\_map pm  , tbl\_pcurl\_info p  , tbl\_pcurl\_display\_map d  , tbl\_user\_info u  where g.urlgrp\_seq=pm.urlgrp\_seq  and pm.url\_seq=p.url\_seq  and g.urlgrp\_seq=d.urlgrp\_seq  and (case when g.target\_type=1 then u.user\_seq=d.target\_seq  when g.target\_type=2 then u.dept\_seq=d.target\_seq  when g.target\_type=0 then g.target\_type=g.target\_type  end  )  -- 정책에 따라서 차단된 URL 리스트  union all  select 1 , 0 , g.urlgrp\_seq , g.urlgrp\_name , g.urlgrp\_desc , p.allow\_flag , p.user\_seq  from tbl\_pcurl\_grp g  , view\_pcurl\_user\_policy p  , tbl\_user\_info u  where g.urlgrp\_seq=p.urlgrp\_seq  and p.allow\_flag=''2''  and p.user\_seq=u.user\_seq  union all  select 0, g.urlgrp\_seq, g.url\_seq, g.url, g.url\_desc, p.allow\_flag , u.user\_seq  from view\_pcurl\_info g  , view\_pcurl\_user\_policy p  , tbl\_user\_info u  where g.urlgrp\_seq=p.urlgrp\_seq  and p.allow\_flag=''2''  and p.user\_seq=u.user\_seq  ) a  where a.user\_seq='||i\_userseq||' --201905221700000001  order by lockstatus asc, a.p\_seq, a.url, a.u\_seq  ';  RAISE NOTICE 'QUANTITY HERE IS %', V\_SQL; -- PRINTS 50  RETURN QUERY EXECUTE V\_SQL;  END;  $BODY$;  ALTER FUNCTION public.func\_nl2\_urluserassignemt(bigint)  OWNER TO hsck; | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL 관리(PUM) | URL 리스트 (UL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUM.UL** | **기능번호** | **SD.PUM.3** |
| **기능설명** | 승인된 URL 그룹 및 URL 리스트를 조회한다. | | |
| **입력** | 승인된 URL 그룹 및 URL 리스트 조회 요청. | | |
| **출력** | 승인된 URL 그룹 및 URL 리스트 | | |
| \* Step 1: URL 관리 조회 화면에서 URL List 버튼 클릭 시 User Sequence 정보를 사용하여  서버 DB 조회를 통해 사용 승인된 URL 그룹 및 URL 리스트 정보를 얻어온다.  \* Step 2: 서버로부터 수신받은 승인된 URL 그룹 및 URL 리스트 정보는 사용 URL 목록 화면  팝업 시 트리에 그룹별로 표출이 되고 사용자가 그룹 아이템을 클릭하면  그룹에 속한 URL 리스트 정보를 확인한다.  Cf) 사용 승인된 URL 그룹 및 URL 리스트 DB 조회 쿼리   |  | | --- | | CREATE OR REPLACE FUNCTION public.func\_nl2\_urluserlist(  i\_userseq bigint)  RETURNS TABLE(o\_grptype character, o\_pseq bigint, o\_urlseq bigint, o\_urlname character varying, o\_urldesc character varying, o\_lockstatus character)  LANGUAGE 'plpgsql'  COST 100  VOLATILE  ROWS 1000  AS $BODY$  DECLARE  V\_SQL varchar; -- 쿼리  WHAND VARCHAR; -- 검색조건이 있을때의 연결(WHERE OR AND)  BEGIN  V\_SQL:='  select distinct  cast(gtype as character),  cast(p\_seq as bigint),  cast(u\_seq as bigint),  cast(url as character varying),  cast(case when replace(a.urldesc, '' '', '''')=''''then ''-'' else a.urldesc end as character varying),  cast(lockstatus as character)  from (  select 1 gtype, 0 p\_seq, g.urlgrp\_seq u\_seq, g.urlgrp\_name url, coalesce(g.urlgrp\_desc, ''-'') urldesc, p.allow\_flag lockstatus, p.user\_seq  from tbl\_pcurl\_grp g  , view\_pcurl\_user\_policy p  , tbl\_user\_info u  where g.urlgrp\_seq=p.urlgrp\_seq  and p.user\_seq=u.user\_seq  union all  select 0, g.urlgrp\_seq, g.url\_seq, g.url, coalesce(g.url\_desc, ''-''), p.allow\_flag , u.user\_seq  from view\_pcurl\_info g  , view\_pcurl\_user\_policy p  , tbl\_user\_info u  where g.urlgrp\_seq=p.urlgrp\_seq  and p.user\_seq=u.user\_seq  ) a  where user\_seq='||i\_userseq||'--201905221700000001  order by lockstatus asc, p\_seq, url  ;  ';  RAISE NOTICE 'QUANTITY HERE IS func\_nl2\_urluserlist %', V\_SQL; -- PRINTS 50  RETURN QUERY EXECUTE  V\_SQL;  END;  $BODY$;  ALTER FUNCTION public.func\_nl2\_urluserlist(bigint)  OWNER TO hsck; | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL 관리(PUM) | 요청사유(URC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUM.URC** | **기능번호** | **SD.PUM.4** |
| **기능설명** | URL 관리 조회 화면에서 요청사유를 입력하여 사용 신청 URL 현황을 조회한다. | | |
| **입력** | URL 사용 신청 현황 조회 요청. | | |
| **출력** | URL 사용 신청 현황. | | |
| \* Step 1: URL 관리 조회 화면에서 사용자가 요청사유를 입력하고 조회 버튼을 클릭한다.  \* Step 2: 조회 버튼 클릭 후 서버 DB 조회를 통해 요청사유가 일치하는  URL 사용 신청 현황 조회 응답을 서버로부터 수신 받는다.  \* Step 3: 서버로부터 수신받은 응답 데이터를 파싱한 후 URL 관리 조회 화면에 조회 결과를  표출한다.  Cf) URL 사용 신청현황 DB 조회 쿼리 – 문서 기능번호 SD.PUM.1 참조. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL 결재(PUA) | URL 조회(URL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUA.URL** | **기능번호** | **SD.PUA.1** |
| **기능설명** | URL 결재 조회 화면에서 URL 을 입력하여 결재요청된 URL 사용신청 현황을 조회한다. | | |
| **입력** | 결재요청된 URL 사용신청 현황 조회 요청. | | |
| **출력** | 결재요청된 URL 사용신청 현황. | | |
| \* Step 1: URL 결재 조회 화면에서 사용자가 검색하려는 URL을 입력하고 조회 버튼을 클릭한다.  \* Step 2: 조회 버튼 클릭 후 서버 DB 조회를 통해 대표 URL과 일치하는  결재요청된 URL 신청 현황 조회 응답을 서버로부터 수신 받는다.  \* Step 3: 서버로부터 수신받은 응답 데이터를 파싱한 후 URL 결재 조회 화면에 조회 결과를  표출한다.  Cf) 결재요청된 URL 사용 신청 현황 조회 쿼리   |  | | --- | | CREATE OR REPLACE FUNCTION public.func\_nl2\_urlapprovesearch(  i\_userseq bigint,  i\_fromdate character varying,  i\_todate character varying,  i\_approvestatus character,  i\_requsername character varying,  i\_url character varying,  i\_reqreason character varying,  i\_sql integer)  RETURNS TABLE(o\_transseq bigint, o\_approveseq bigint, o\_approvepossible character, o\_approvestatus2 character varying, o\_approvestatus character varying, o\_requsername character varying, o\_title character varying, o\_titleurl character varying, o\_reqdate character varying, o\_arpprovedate character varying)  LANGUAGE 'plpgsql'  COST 100  VOLATILE  ROWS 1000  AS $BODY$  DECLARE  V\_SQL varchar; -- 쿼리  WHAND VARCHAR; -- 검색조건이 있을때의 연결(WHERE OR AND)  BEGIN  V\_SQL:='  SELECT  A.TRANS\_SEQ  , A.REQ\_SEQ  , CAST((CASE WHEN GETAPPROVEPOSSIBLE2(A.APPROVE\_USER\_SEQ, A.TRANS\_SEQ, A.APPROVE\_ORDER) > 0 THEN 1 ELSE 0 END) AS CHARACTER) APPROVEPOSSIBLE  , CAST(A.D\_POS AS VARCHAR ) APPROVESTAUTS2  , CAST ((CASE WHEN P.REQ\_FLAG =''2'' THEN ''11'' ELSE A.APPROVE\_FLAG END) AS CHARACTER VARYING) APPROVESTAUTS  , CAST (U.USER\_NAME || (CASE WHEN COALESCE(U.USER\_RANK, U.USER\_POSITION, '''') !='' '' THEN ('' ''||COALESCE(U.USER\_RANK, U.USER\_POSITION) ) ELSE '''' END) AS VARCHAR)  , P.TITLE  , P.CONTENT  , CAST(P.REQUEST\_TIME AS CHARACTER VARYING)  , CAST(P.APPROVE\_TIME AS CHARACTER VARYING)  FROM  (  SELECT \*  FROM TBL\_PCURL\_REQ\_INFO  UNION ALL  SELECT \*  FROM TBL\_PCURL\_REQ\_HIS  WHERE REQUEST\_TIME BETWEEN '''||i\_fromdate||''' AND '''||i\_todate||'''  ) P  INNER JOIN (  SELECT 1 D\_POS, \* FROM TBL\_APPROVE\_INFO  UNION ALL  SELECT 0 D\_POS, \* FROM TBL\_APPROVE\_HIS  WHERE APPR\_REQ\_TIME BETWEEN '''||i\_fromdate||''' AND '''||i\_todate||'''  ) A ON P.TRANS\_SEQ = A.TRANS\_SEQ  LEFT OUTER JOIN  (  SELECT \*  FROM TBL\_USER\_SFM S  WHERE NOW() BETWEEN TO\_DATE(FROMDATE||''000000'', ''YYYYMMDDHH24MISS'') AND TO\_DATE(TODATE||''235959'', ''YYYYMMDDHH24MISS'')  ) S ON A.APPROVE\_USER\_SEQ=S.USER\_SEQ  LEFT JOIN TBL\_USER\_INFO U ON U.USER\_SEQ=A.USER\_SEQ  WHERE (A.APPROVE\_USER\_SEQ='||i\_userseq||' OR S.SFM\_USER\_SEQ='||i\_userseq||')  ';    -- i\_approvestatus  -- 결재상태  IF i\_approvestatus IS NOT NULL AND i\_approvestatus!='' AND i\_approvestatus!='0' THEN  -- V\_SQL:=V\_SQL||CHR(13)||'AND (P.REQ\_FLAG =''1'' AND P.APPROVE\_FLAG='''||i\_approvestatus||''')';  -- 1: 요청, 2: 취소  V\_SQL:=V\_SQL||CHR(13)||'AND (case when p.req\_flag =''2'' then ''11'' else a.approve\_flag end) = '''||i\_approvestatus||'''';  END IF;      IF i\_requsername IS NOT NULL AND i\_requsername!='' THEN  V\_SQL:=V\_SQL||CHR(13)||'AND u.user\_name like ''%'|| i\_requsername ||'%''';  END IF;    -- 포함URL  -- character  IF i\_url IS NOT NULL AND i\_url!='' THEN  V\_SQL:=V\_SQL||CHR(13)||'AND func\_nl2\_urlcount(p.trans\_seq, '''||i\_url||''')>0';  END IF;  -- 사유  IF i\_reqreason IS NOT NULL AND i\_reqreason!='' THEN  V\_SQL:=V\_SQL||CHR(13)||'AND title like ''%'||i\_reqreason||'%''';  END IF;  V\_SQL:=V\_SQL||CHR(13)||'order by request\_time';      IF i\_sql = 1 THEN  RAISE NOTICE 'QUANTITY HERE IS %', V\_SQL; -- PRINTS 50  END IF;  RETURN QUERY EXECUTE  V\_SQL;  END;  $BODY$;  ALTER FUNCTION public.func\_nl2\_urlapprovesearch(bigint, character varying, character varying, character, character varying, character varying, character varying, integer)  OWNER TO hsck; | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL 결재(PUA) | 요청사유(RC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUA.RC** | **기능번호** | **SD.PUA.2** |
| **기능설명** | URL 결재 조회 화면에서 요청사유를 입력하여 결재요청된 URL 사용신청 현황을 조회한다. | | |
| **입력** | 결재요청된 URL 사용신청 현황 조회 요청. | | |
| **출력** | 결재요청된 URL 사용신청 현황 리스트 | | |
| \* Step 1: URL 결재 조회 화면에서 사용자가 검색하려는 요청사유를 입력하고 조회 버튼을 클릭한다.  \* Step 2: 조회 버튼 클릭 후 서버 DB 조회를 통해 요청사유가 일치하는  결재요청된 URL 사용신청 현황 조회 응답을 서버로부터 수신 받는다.  \* Step 3: 서버로부터 수신받은 응답 데이터를 파싱한 후 URL 결재 조회 화면에 조회 결과를  표출한다.  Cf) 결재요청된 URL 사용신청 현황 조회 쿼리 – 문서 기능번호 SD.PUA.1 참조. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL상세보기(PUD) | 적용부서명(DN) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUD.DN** | **기능번호** | **SD.PUD.1** |
| **기능설명** | 신청된 URL 을 승인 시 적용될 부서명 정보를 나타낸다. | | |
| **입력** | PCURL 상세보기 요청. | | |
| **출력** | 사용 가능한 부서명 정보. | | |
| \* Step 1: Agent 는 사용자가 조회 화면에서 상세보기 버튼 클릭 시 User Sequence 정보와  신청 URL Sequence 정보를 사용하여 서버 DB 조회를 통해 상세보기에 표현될 정보를  얻어온다.  \* Step 2: 상세보기에 관련된 정보를 서버로부터 수신받은 후 수신 받은 데이터를 파싱하여  승인 후 적용될 부서명 정보를 얻어온다.  \* Step 3: 로드된 적용 부서 정보는 상세보기 창의 오른쪽 맨위 에디트 창에 표시되며  적용 부서 정보가 존재하지 않으면 빈 값(‘-“)으로 표시된다.  Cf) URL 신청 상세보기 DB 조회 쿼리   |  | | --- | | CREATE OR REPLACE FUNCTION func\_nl\_afterapproveperiod(  IN i\_systemid character varying,  IN i\_approve\_user\_seq bigint)  RETURNS TABLE(o\_approveafter character varying, o\_dept\_seq bigint, o\_dept\_name character varying, o\_cnt integer) AS  $BODY$  DECLARE  V\_AAMC VARCHAR;  V\_SQL VARCHAR; -- 쿼리  BEGIN  /\*\*  input:  I\_SYSTEMID : 시스템 아이디  I\_APPROVE\_USER\_SEQ : 결재자 시퀀스  output:  O\_APPROVEAFTER : 사후결재건수체크여부  O\_DEPT\_SEQ : 결재자의 부서 시퀀스  O\_DEPT\_NAME : 결재자의 부서명  O\_CNT : 사후결재 승인대기 건수  \*/  V\_AAMC := (SELECT COALESCE(MAX(TAG\_VALUE), '0')  FROM TBL\_SYSTEM\_ENV  WHERE TAG='AFTERAPPROVEMAXPERIOD'  AND SUBSTRING(SYSTEM\_ID, 1, 1)=I\_SYSTEMID  GROUP BY SUBSTRING(SYSTEM\_ID, 1, 1)  );  V\_AAMC:=COALESCE(V\_AAMC, '0');    IF V\_AAMC ='0' THEN  V\_SQL:='SELECT CAST(''0'' AS VARCHAR) OAA, CAST(0 AS BIGINT) OSEQ, CAST(''-'' AS VARCHAR) ONAME, CAST(0 AS INT) OCNT';  ELSE  V\_SQL:='  SELECT CAST('||V\_AAMC||' AS VARCHAR) OAA, CAST(DEPT\_SEQ AS BIGINT) OSEQ, CAST(DEPT\_NAME AS VARCHAR) ONAME, CAST(COUNT(\*) AS INT) OCNT  FROM (  SELECT DISTINCT D.DEPT\_SEQ, D.DEPT\_NAME  , A.TRANS\_SEQ  , A.USER\_SEQ  FROM TBL\_APPROVE\_AFTER A  , TBL\_USER\_INFO U  , TBL\_USER\_INFO DU  , TBL\_DEPT\_INFO D  WHERE A.APPROVE\_USER\_SEQ=U.USER\_SEQ  AND DU.DEPT\_SEQ=D.DEPT\_SEQ  AND DU.DEPT\_SEQ=U.DEPT\_SEQ  AND COALESCE(U.ACCOUNT\_EXPIRES, ''99991231'') > TO\_CHAR(NOW(), ''YYYYMMDD'')  AND U.USE\_STATUS = ''1''  AND To\_DATE(A.appr\_Req\_time, ''YYYYMMDDHH24miss'') < (current\_date-CAST('||V\_AAMC||' AS INT))  AND DU.USER\_SEQ='||I\_APPROVE\_USER\_SEQ||'  ) A  GROUP BY DEPT\_SEQ, DEPT\_NAME  ORDER BY DEPT\_SEQ';    END IF ;    RAISE NOTICE 'BY SQL IS %', V\_SQL; -- PRINTS 50  --201705151700000026    RETURN QUERY EXECUTE  V\_SQL;  END;  $BODY$  LANGUAGE plpgsql VOLATILE  COST 100  ROWS 1000;  ALTER FUNCTION func\_nl\_afterapproveperiod(character varying, bigint)  OWNER TO hsck; | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL상세보기(PUD) | 요청사유(RTR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUD.RTR** | **기능번호** | **SD.PUD.2** |
| **기능설명** | 사용자가 URL 사용 신청 시 입력한 신청 사유 정보를 나타낸다. | | |
| **입력** | PCURL 상세보기 요청. | | |
| **출력** | 신청 사유 | | |
| \* Step 1: Agent 는 사용자가 조회 화면에서 상세보기 버튼 클릭 시 User Sequence 정보와  신청 URL Sequence 정보를 사용하여 서버 DB 조회를 통해 상세보기에 표현될 정보를  얻어온다.  \* Step 2: 상세보기에 관련된 정보를 서버로부터 수신받은 후 수신 받은 데이터를 파싱하여  사용자가 URL 신청 시 입력한 신청 사유 정보를 얻어온다..  \* Step 3: 로드된 신청 사유 정보는 상세보기 창의 요청사유 에디트 창에 표시 된다.  Cf) URL 신청 상세보기 DB 조회 쿼리 – 문서 기능번호 SU.PUD.1 참조. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL상세보기(PUD) | URL 신청 현황(URD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUD.URD** | **기능번호** | **SD.PUD.3** |
| **기능설명** | 사용자가 신청한 URL 리스트를 보여준다. | | |
| **입력** | PCURL 상세보기 요청. | | |
| **출력** | 사용 신청한 URL 리스트 | | |
| \* Step 1: Agent 는 사용자가 조회 화면에서 상세보기 버튼 클릭 시 User Sequence 정보와  신청 URL Sequence 정보를 사용하여 서버 DB 조회를 통해 상세보기에 표현될 정보를  얻어온다.  \* Step 2: 상세보기에 관련된 정보를 서버로부터 수신받은 후 수신 받은 데이터를 파싱하여  사용자가 URL 신청 시 신청한 URL 리스트를 얻어온다  \* Step 3: 로드된 신청 URL 리스트는 상세보기창의 URL 리스트 창에 표시된다.  Cf) URL 신청 상세보기 DB 조회 쿼리 – 문서 기능번호 SU.PUD.1 참조. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL상세보기(PUD) | 결재 내역(AL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUD.AL** | **기능번호** | **SD.PUD.4** |
| **기능설명** | URL 사용 신청 건에 대해 결재자 결재 상태 정보(승인이력 정보)를 표시한다. | | |
| **입력** | PCURL 상세보기 요청. | | |
| **출력** | 결재자 결재 상태(승인이력). | | |
| \* Step 1: Agent 는 사용자가 조회 화면에서 상세보기 버튼 클릭 시 User Sequence 정보와  신청 URL Sequence 정보를 사용하여 서버 DB 조회를 통해 상세보기에 표현될 정보를  얻어온다.  \* Step 2: 상세보기에 관련된 정보를 서버로부터 수신받은 후 수신 받은 데이터를 파싱하여  결재자 결재상태정보 (승인이력) 를 얻어온다.  \* Step 3: 로드된 승인이력 정보 확인은 결재내역 버튼을 클릭하면 승인이력을 표시하는 리스트  창이 팝업된다.  \* Step 4. 승인이력 정보를 표출할 결재내역 버튼은 결재 방법에 따라 보여지는데 조건은  AND 결재 사용 시 결재단계가 2단계 이상이고 OR or AND/OR 결재 사용 시일때만  버튼이 표출된다.  Cf) URL 신청 상세보기 DB 조회 쿼리 – 문서 기능번호 SU.PUD.1 참조. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | PCURL상세보기(PUD) | 결재 정보(AI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.PUD.AI** | **기능번호** | **SD.PUD.5** |
| **기능설명** | URL 사용 신청 건에 대해 결재단계의 마지막 정보를 표시한다. | | |
| **입력** | PCURL 상세보기 요청. | | |
| **출력** | 결재 마지막 단계의 결재 상태. | | |
| \* Step 1: Agent 는 사용자가 조회 화면에서 상세보기 버튼 클릭 시 User Sequence 정보와  신청 URL Sequence 정보를 사용하여 서버 DB 조회를 통해 상세보기에 표현될 정보를  얻어온다.  \* Step 2: 상세보기에 관련된 정보를 서버로부터 수신받은 후 수신 받은 데이터를 파싱하여  결재 마지막상태 정보를 얻어온다.  \* Step 3: 로드된 결재 마지막상태 정보 확인은 상세보기 창 맨 아래 영역에 표시되며  결재자, 결재상태, 결재일 , 반려사유가 표시된다.  Cf ) 결재자 – 결재자 이름, 결재상태 – 승인대기, 승인, 반려  결재일 – 결재자가 결재(승인 또는 반려)를 수행한 날짜  반려사유 – 결재자가 반려를 수행하였을 경우에만 표시되며 나머지 경우는 빈 값으로 표시.  \* Step 4. 결재 마지막상태 정보는 결재 방법에 따라 다르게 보여진다.  - AND 결재 사용 시 결재단계가 1단계 일 경우 1단계 결재자 결재 상태 표시  - AND 결재 사용 시 결재단계가 2단계 이상일 경우 전단계 결재 상태 조건에 따라 다르게 표시.  - OR 결재 사용 시 모든 결재자가 결재 이전 상태이면 처음 결재자를 표시하고  한 명의 결재자가 결재를 수행하면 결재를 수행한 결재자 정보를 표시.  - AND/OR 결재 사용 시 AND 결재자 표시 조건과 OR 결재자 표시 조건을 복합적으로  확인하고 표시.  Cf) URL 신청 상세보기 DB 조회 쿼리 – 문서 기능번호 SU.PUD.1 참조. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | 클립보드 관리(CM) | 데이터 타입 (DT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.CM.DT** | **기능번호** | **SD.CM.1** |
| **기능설명** | 클립보드 관리 조회 화면에서 클립보드 데이터 타입에 따라 조회한다. | | |
| **입력** | 클립보드 데이터 타입 | | |
| **출력** | 클립보드 데이터 타입에 따라 조회된 결과. | | |
| \* Step 1: 클립보드 관리 조회 화면으로 이동 후 오른쪽 상단의 데이터타입 콤보박스를 클릭한다.  \* Step 2: 데이터타입 콤보박스에서 드랍되어 표출되는 타입 중 한 개를 선택하고 조회 버튼을  클립한다.  Cf) 콤보박스에서 드랍되는 클립보드 데이터타입 : 전체, 텍스트, 이미지  \* Step 3: 조회 버튼을 클릭하면 프로토콜 FILE\_TRANSLIST 를 통해 조회를 요청한다.  사용자가 선택한 클립보드 데이터 타입은 DATATYPE 태그 값으로 변환되어 조회를  요청한다.  Cf) DATATYPE 태그 값 : 0 or 값이 없을 때 – 일반파일 조회  1 – 클립보드 텍스트 ( txt )  2 – 클립보드 이미지 ( bmp )  3 – 클립보드 전체 ( txt + bmp )  \* Step 4: 서버로부터 조회 응답을 수신받으면 클립보드 관리 조회 화면에 결과값을 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 조회(SD) | 클립보드 결재(CA) | 데이터 타입 (DT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **SD.CA.DT** | **기능번호** | **SD.CA.1** |
| **기능설명** | 클립보드 결재 조회 화면에서 클립보드 데이터 타입에 따라 조회한다. | | |
| **입력** | 클립보드 데이터 타입 | | |
| **출력** | 클립보드 데이터 타입에 따라 조회된 결과. | | |
| \* Step 1: 클립보드 결재 조회 화면으로 이동 후 오른쪽 상단의 데이터타입 콤보박스를 클릭한다.  \* Step 2: 데이터타입 콤보박스에서 드랍되어 표출되는 타입 중 한 개를 선택하고 조회 버튼을  클립한다.  Cf) 콤보박스에서 드랍되는 클립보드 데이터타입 : 전체, 텍스트, 이미지  \* Step 3: 조회 버튼을 클릭하면 프로토콜 FILE\_APPROVE 를 통해 조회를 요청한다.  사용자가 선택한 클립보드 데이터 타입은 DATATYPE 태그 값으로 변환되어 조회를  요청한다.  Cf) DATATYPE 태그 값 : 0 or 값이 없을 때 – 일반파일 조회  1 – 클립보드 텍스트 ( txt )  2 – 클립보드 이미지 ( bmp )  3 – 클립보드 전체 ( txt + bmp )  \* Step 4: 서버로부터 조회 응답을 수신받으면 클립보드 결재 조회 화면에 결과값을 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재라인(APL) | 기본 결재라인 (SAL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APL.SAL** | **기능번호** | **AP.APL.1** |
| **기능설명** | 결재를 받을 기본 결재라인을 서버로부터 받아옴. | | |
| **입력** |  | | |
| **출력** | 결재라인 | | |
| \* Step 1: Agent가 서버로 $APPROVEDEFAULT 전문을 보내 결재라인을 요청.  \* Step 2: 서버로부터 $APPROVEDEFAULT\_ACK로 정보를 받음.  \* Step 3: 결재라인 정보는 본인, 결재자순으로 정보가 있음..  \* Step 4: 본인을 제외한 결재자 순으로 결재라인을 생성함. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재라인(APL) | 결재자 선택 (ARS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APL.ARS** | **기능번호** | **AP.APL.2** |
| **기능설명** | 기본 결재라인을 사용자가 변경하는 기능  검색(이름 / 부서), 조직도상에서 선택, 검색(이름 / 부서)과 조직도상에서 선택 혼합 방식으로 3가지를 제공하며 결재자들의 결재순서를 조정가능함. | | |
| **입력** | 부서명, 부서Seq, 이름 | | |
| **출력** | 결재라인 | | |
| 검색(이름 / 부서) <입력 : 부서명, 이름>  \* Step 1: 사용자가 부서명이나 이름을 입력함  \* Step 2: $HSCKQUERY전문을 통하여 DB에서 연관 사용자를 조회함  \* Step 3: 조회된 사용자는 $HSCKQUERY\_ACK를 통해 받음  \* Step 4: 조회된 사용자를 리스트에 표시함  \* Step 5: 리스트에 표시된 사용자 중 결재자를 선택하여 결재라인 생성함  조직도상에서 선택 <입력 : 부서Seq>  \* Step 1: 본인이 포함된 조직(부서) 정보를 $DEPTINFO 전문을 통해 요청함  \* Step 2: $DEPTINFO\_ACK로 받은 조직정보를 Tree 행태로 표시함  \* Step 3: 사용자가 Tree의 조직을 선택하여 $USERINFO2전문을 통해 사용자 조회를 요청  \* Step 4: $USERINFO2\_ACK로 결재자 정보를 받아 조회된 사용자를 리스트에 표시함  \* Step 5: 리스트에 표시된 사용자 중 결재자를 선택하여 결재라인 생성함 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재라인(APL) | 복합결재 (CA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APL.CA** | **기능번호** | **AP.APL.3** |
| **기능설명** | OR 결재, ANDOR 결재. | | |
| **입력** |  | | |
| **출력** | 결재라인 | | |
| 결재 방식은 AND, OR, ANDOR 종류가 있음  AND 결재 : 결재자의 순차적 결재 방식 (기본방식)  OR 결재 : 결재자 중 1명이라도 결재를 하면 결재가 완료되는 방식  ANDOR 결재 : 결재자 그룹 단위로 결재 가지며, 그룹 중 한명이라도 결재를 하면 다음 그룹으로 결재가 넘어가는 방식  AND 결재라인  \* Step 1: 기능코드 AP.APL.SAL, AP.APL.ARS 에서 결재라인을 결재자별 (,)로 구분하여보관  예) 결재자seq,결재자seq,결재자seq  OR 결재라인  \* Step 1: 기능코드 AP.APL.SAL, AP.APL.ARS 에서 결재라인을 결재자별 (|)로 구분하여보관  예) 결재자seq|결재자seq|결재자seq  ANDOR 결재라인  \* Step 1: 기능코드 AP.APL.SAL, AP.APL.ARS 에서 결재라인을 결재자별 (|)로 구분과 그룹별 (,)로 구분하여보관  예) 결재자seq|결재자seq|결재자seq,결재자seq|결재자seq|결재자seq  \* Step 2: (공통) 파일 전송시 $TRANSREQ 전문에 보관된 결재라인을 포함함(ConfirmId). | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재라인(APL) | 타사시스템 결재라인연동. (ALW) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APL.ALW** | **기능번호** | **AP.APL.4** |
| **기능설명** | 정보보호포털 및 EAI 결재라인 연동. | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재라인(APL) | 결재라인 저장 (ALS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APL.ALS** | **기능번호** | **AP.APL.5** |
| **기능설명** | 로컬PC에 마직막으로 사용된 결재라인 저장. (AND, OR, AND/OR 결재라인 지원) | | |
| **입력** |  | | |
| **출력** | 결재라인 파일 | | |
| 사용자가 선택한 결재라인을 옵션에 따라 로컬 PC에 저장하여, 다시 로그인을 했을 경우 저장된 결재라인을 사용하는 기능  \* Step 1: 기능코드 AP.APL.SAL, AP.APL.ARS 에서 결재라인 선택  \* Step 2: 선택된 결재라인의 정보를 암호화하여 로컬PC에 파일로 저장  결재라인정보 : 버전, 저장일시, 아이디, 결재단계타입(AND, OR, AND/OR), 결재수  결재정보(결재단계 사용자아이디 사용자이름 사용자org이름 사용자팀명 사용자seq 사용자직책 사용자직위)  \* Step 3: 재 로그인시 저장된 파일 복호화  \* Step 4: 복호화된 정보의 아이디 및 저장일시를 확인하여 결재라인 갱신  \* Step 5: 복호화된 정보의 아이디가 다를 경우 파일 파기 및 결재라인 변경 안함  \* Step 6: 복호화된 정보의 저장일시가 저장기간을 초과 할 경우 파일 파기 및 결재라인 변경 안함(저장기간이 옵션이 있을 경우) | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재라인(APL) | 보안결재 (SA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APL.SA** | **기능번호** | **AP.APL.6** |
| **기능설명** | 전송 건의 개인정보 검출 유무에 따라 보안결재 사용 | | |
| **입력** |  | | |
| **출력** |  | | |
| 전송 건의 개인정보 검출 유무에 따라 보안결재를 추가하는 기능  개인정보 검출은 연동 위치에 따라 Agent/Server로 기능이 나뉨  Agent개인정보 연동  \* Step 1: 사용자가 전송할 파일을 선택하여 전송 리스트에 추가  \* Step 2: 개인정보 연동을 통해 개인정보 유무 판별 및 표시  \* Step 3: 개인정보가 포함되어 있을 경우 파일전송시 전문 $Transreq의 DLP 테그에 1로 설정  FileRecord 에 DLP 1과 DLPCONTENT 개인정보내용 설정  \* Step 4: 서버 설정에 보안결재가 진행  Server개인정보 연동  \* Step 1: 사용자가 전송할 파일을 선택하여 전송 및 서버에서 개인정보 검출  \* Step 2: 개인정보가 있을 경우 옵션에 따라 Agent에 전문 $PRIVACY\_NOTIFY로 Agent에 개인정보 알림  \* Step 3: 서버에 전문 $PRIVACY\_CONTINUE로 결재진행여부 (DLPAPPROVE) 응답으로 보냄  \* Step 4: 응답 받은 서버는 보안결재진행이나 전송취소를 함 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재라인(APL) | 특정조건 보안결재 사용 (SSAU) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APL.SSAU** | **기능번호** | **AP.APL.7** |
| **기능설명** | 전송 건의 개인정보 검출 건수에 따라 보안결재 사용. | | |
| **입력** |  | | |
| **출력** |  | | |
| 전송 건의 개인정보 검출 유무에 따라 보안결재를 추가하는 기능  개인정보 검출은 연동 위치에 따라 Agent/Server로 기능이 나뉨  기능코드 AP.APL.SA와 달리 개인정보가 포함되더라도 개인정보 개수에 따라 개인정보 포함표시 달리함  Agent개인정보 연동  \* Step 1: 사용자가 전송할 파일을 선택하여 전송 리스트에 추가  \* Step 2: 개인정보 연동을 통해 개인정보 유무 판별 및 표시  \* Step 3: 서버로 파일을 전송시 개인정보 개수를 확인하여 옵션 값이 이상일 경우만 전문 $Transreq의 DLP 테그에 1로 설정  \* Step 4: 개인정보가 포함되어 있을 경우 파일전송시 전문 $Transreq의 FileRecord 에 DLP 1과 DLPCONTENT 개인정보내용 설정  \* Step 5: 서버 설정에 보안결재가 진행  Server개인정보 연동  \* Step 1: 사용자가 전송할 파일을 선택하여 전송 및 서버에서 개인정보 검출  \* Step 2: 개인정보가 있을 경우 개인정보 검출 건수 옵션에 따라 Agent에 전문 $PRIVACY\_NOTIFY로 Agent에 개인정보 알림  \* Step 3: 서버에 전문 $PRIVACY\_CONTINUE로 결재진행여부 (DLPAPPROVE) 응답으로 보냄  \* Step 4: 응답 받은 서버는 보안결재진행이나 전송취소를 함 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재라인(APL) | 보안결재자 사용자 선택 (SAUS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APL.SAUS** | **기능번호** | **AP.APL.8** |
| **기능설명** | 개인정보 검출 시 사용자가 보안결재자 선택. | | |
| **입력** |  | | |
| **출력** |  | | |
| 전송 건의 개인정보 검출 유무에 따라 보안결재를 추가하는 기능  개인정보 검출은 연동 위치에 따라 Agent/Server로 기능이 나뉨  설정에 따라 Server에서 개인정보를 검출하였을 때 보안결재자를 사용자가 선택하는 기능  \* Step 1: 개인정보 검출에 대한 사용자에게 Server가 전문 $PRIVACY\_NOTIFY를 보냄  \* Step 2: 사용자는 보안결재자를 기능코드 AP.APL.SAL, AP.APL.ARS 에서 조회 및 선택  \* Step 3: 서버에 전문 $PRIVACY\_CONTINUE 로 DLPAPPROVE(결재진행여부)와 PRIVACYCONFIRMID(결재라인을 응답으로 보냄  \* Step 4: 응답 받은 서버는 보안결재진행이나 전송취소를 함   * 참고 PRIVACYCONFIRMID 결재라인은 기능코드 AP.APL.CA 와 동일한 형식 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 대결재 (API) | 일반 대결재 선택 (SIAS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.API.SIAS** | **기능번호** | **AP.API.1** |
| **기능설명** | 검색 / 선택해서 다른 결재자에게 일정기간동안 대결재 권한 부여. | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재자는 다른 결재자 및 일반 사용자를 선택하여 결재를 대신 할 수 있도록 할 수 있음  \* Step 1: 결재자는 기능코드 AP.APL.SAL, AP.APL.ARS 에서 조회 및 단일선택(일반 및 결재자)  \* Step 2: 결재자는 전문 $APPR\_INST\_REG 로 대결재자 정보를 전송  \* Step 3: 서버 설정 APPROVETYPESFM에 따라 동작함  APPROVETYPESFM 1이면 사용자의 기본결재자가 대결재자로 변경되어 진행(재로그인시)  APPROVETYPESFM 2이면 사용자의 기본결재자로 결재를 올려면 기본결재자와 대결재자가 결재를 진행   * 참고   서버 설정 APPROVEPROXYRIGHT 1 이면 일반 결재자 이상만 조회  서버 설정 APPROVEPROXYRIGHT 2 이면 일반 사용자도 조회 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 대결재 (API) | 다중 대결재 선택 (MIAS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.API.MIAS** | **기능번호** | **AP.API.2** |
| **기능설명** | 검색 / 선택해서 여러명의 결재자에게 일정기간동안 대결재 권한 부여. | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재자는 다른 결재자 및 일반 사용자를 선택하여 결재를 대신 할 수 있도록 할 수 있음  기능코드 AP.API.SIAS 와 달리 여러명의 대결재자를 선택하는 기능  \* Step 1: 결재자는 기능코드 AP.APL.SAL, AP.APL.ARS 에서 조회 및 다중선택(일반 및 결재자)  \* Step 2: 결재자는 전문 $HSCKQUERY로 대결재자 정보를 DB tbl\_user\_sfm에 등록  \* Step 3: 서버 설정 APPROVETYPESFM에 따라 동작함  APPROVETYPESFM 1이면 사용자의 기본결재자가 대결재자로 변경되어 진행(재로그인시)  APPROVETYPESFM 2이면 사용자의 기본결재자로 결재를 올려면 기본결재자와 대결재자들이 결재를 진행   * 참고   서버 설정 APPROVEPROXYRIGHT 1 이면 일반 결재자 이상만 조회  서버 설정 APPROVEPROXYRIGHT 2 이면 일반 사용자도 조회 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 대결재 (API) | 대결재 수정 (IADUR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.API.IADUR** | **기능번호** | **AP.API.3** |
| **기능설명** | 이미 지정된 대결재자에 대해 대결재 기간 및 대결재자 수정. | | |
| **입력** |  | | |
| **출력** |  | | |
| 일반 대결재 기능일 결우  \* Step 1: 결재자는 전문 $APPR\_INST\_CUR 대결재자 리스트를 조회 함  \* Step 2: 대결재자가 존재 할 경우 기간 및 대결재자를 변경하여 전문 $APPR\_INST\_REG로 대결재자 정보를 전송  다중 대결재 기능일 결우  \* Step 1: 결재자는 전문 $HSCKQUERY로 DB tbl\_user\_sfm의 대결재자 리스트를 조회 함  \* Step 2: 대결재자가 존재 할 시 기간 및 대결재자를 변경하여 전문 $HSCKQUERY로 DB tbl\_user\_sfm 대결재자 정보를 갱신 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 대결재 (API) | 대결재 해제 (IAEND) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.API.IAEND** | **기능번호** | **AP.API.4** |
| **기능설명** | 대결재 지정 해제 | | |
| **입력** |  | | |
| **출력** |  | | |
| 일반 대결재 기능일 결우  \* Step 1: 결재자는 전문 $APPR\_INST\_CLEAR로 대결재자를 해제  다중 대결재 기능일 결우  \* Step 1: 결재자는 전문 $HSCKQUERY로 DB tbl\_user\_sfm의 대결재자 리스트를 재거 함 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 대결재 (API) | 대결재 권한상속 (IAR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.API.IAR** | **기능번호** | **AP.API.5** |
| **기능설명** | 대결재 지정시 원결재자의 권한을 대결재자에게 상속. | | |
| **입력** |  | | |
| **출력** |  | | |
| 대결재자로 지정된 사용자는 원결재자들 중 제일 높은 권한을 상속 받음  \* Step 1: 대결재자는 전문 $HSCKQUERY로 DB에서 원결재자들의 정보를 조회함  조회된 정보 결재자ID, 결재자Seq, 결재자명, 결재자직급, ApprPos  \* Step 2: 조회된 정보 중 ApprPos가 제일 높은 원결재자의 권한으로 권한(ApprPos)을 변경함.  본인 보다 직급이 낮을 경우 권한을 변경하지 않음  \* Step 3: 대결재가 해제된 경우 다른 대결재권 중 가장 높은 권한으로 변경  대결재가 없는 경우 원래의 권한으로 돌아감 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재공통 (APC) | 단일 승인 (SA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APC.SA** | **기능번호** | **AP.APC.1** |
| **기능설명** | 결재관련 항목에서 승인 처리하는 기능 | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재관련 승인대기 상태의 건에 대해 승인처리 기능  일반/보안/메일 결재관리 리스트, 상세보기에서 승인가능한 상태를 확인하여 승인을 처리  <일반/보안 결재>  \* Step 1: 결재할 대상이 결재가 가능한 상태인지 확인  승인가능상태 확인 : 승인대기 / 승인가능Flag / 전송상태  \* Step 2: 전문 $APPROVEACTION을 통해 승인처리  Tag ReqSeq : 승인요청 정보Key  Tag Action : **승인(1)**/반려(2)  Tag Reason : 승인사유  Tag ApproveKind : 사전결재(0)/사후결재(1)  Tag APPROVER : 대결재자정보 안남김(0)/대결재자정보 안남김(1)  Tag APPROVEUSERKIND : 일반결재(0)/보안결재(1)  \* Step 3: 전문 $APPROVEACTION\_ACK 서버로 부터 받아 정상 승인처리 확인  <메일 결재>  \* Step 1: 결재할 대상이 결재가 가능한 상태인지 확인  승인가능상태 확인 : 승인대기 / 승인가능Flag / 전송상태  \* Step 2: 전문 $EMAIL\_APPROVE을 통해 승인처리  Tag EmailApproveSeq : 메일전송 요청Key  Tag PROCID: **승인(1)**/반려(2)  Tag Reason : 승인사유  Tag ApproveKind : 사전결재(0)/사후결재(1)  \* Step 3: 전문 $EMAIL\_APPROVE\_ACK 서버로 부터 받아 정상 승인처리 확인 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재공통 (APC) | 일괄 승인 (AA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APC.AA** | **기능번호** | **AP.APC.2** |
| **기능설명** | 결재관련 항목에서 여러건을 일괄적으로 승인 처리하는 기능 | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재관련 승인대기 상태의 건에 대해 일괄적으로 승인처리하는 기능  일반/보안/메일 결재관리 리스트에서 승인가능한 상태를 확인하여 승인을 처리  <일반/보안 결재>  \* Step 1: 결재할 대상이 결재가 가능한 상태인지 확인하여 승인할 목록을 선택  승인가능상태 확인 : 승인대기 / 승인가능Flag / 전송상태  \* Step 2: 전문 $APPROVE\_BATCH를 통해 승인처리  Tag PROCID: **승인(1)**/반려(2)  Tag Reason : 승인사유  Tag ApproveKind : 사전결재(0)/사후결재(1)  Tag APPROVESEQS : 승인요청 정보Key들 (delemiter는 0x02, 최대 100개 제한)  Tag APPROVER : 대결재자정보 안남김(0)/대결재자정보 안남김(1)  Tag APPROVEUSERKIND : 일반결재(0)/보안결재(1)/PCURL결재(2)  \* Step 3: 전문 $APPROVE\_BATCH\_ACK 서버로 부터 받아 정상 승인처리 확인  <메일 결재>  \* Step 1: 결재할 대상이 결재가 가능한 상태인지 확인하여 승인할 목록을 선택  승인가능상태 확인 : 승인대기 / 승인가능Flag / 전송상태  \* Step 2: 전문 $EMAIL\_APPROVE\_BATCH를 통해 승인처리  Tag PROCID: **승인(1)**/반려(2)  Tag Reason : 승인사유  Tag EmailApproveSeqS: 메일전송 요청Key들 (delemiter는 0x02, 최대 100개 제한)  \* Step 3: 전문 $EMAIL\_APPROVE\_BATCH \_ACK 서버로 부터 받아 정상 승인처리 확인 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재공통 (APC) | 반려 (RJ) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APC.RJ** | **기능번호** | **AP.APC.3** |
| **기능설명** | 결재관련 항목에서 반려 처리하는 기능 | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재관련 승인대기 상태의 건에 대해 반려처리 기능 **(사후결재는 기능 미사용)**  일반/보안/메일 결재관리 리스트, 상세보기에서 승인가능한 상태를 확인하여 반려를 처리  <일반/보안 결재>  \* Step 1: 결재할 대상이 결재가 가능한 상태인지 확인  승인가능상태 확인 : 승인대기 / 승인가능Flag / 전송상태  \* Step 2: 전문 $APPROVEREJECT을 통해 반려처리  Tag ReqSeq : 승인요청 정보Key  Tag Action : 승인(1)/**반려(2)**  Tag Reason : 반려사유  Tag APPROVER : 대결재자정보 안남김(0)/대결재자정보 안남김(1)  Tag APPROVEUSERKIND : 일반결재(0)/보안결재(1)  \* Step 3: 전문 $APPROVEREJECT\_ACK 서버로 부터 받아 정상 반려처리 확인  <메일 결재>  \* Step 1: 결재할 대상이 결재가 가능한 상태인지 확인  승인가능상태 확인 : 승인대기 / 승인가능Flag / 전송상태  \* Step 2: 전문 $EMAIL\_APPROVE을 통해 반려처리  Tag EmailApproveSeq : 메일전송 요청Key  Tag PROCID: 승인(1)/**반려(2)**  Tag Reason : 반려사유  Tag ApproveKind : 사전결재(0)/사후결재(1)  \* Step 3: 전문 $EMAIL\_APPROVE\_ACK 서버로 부터 받아 정상 반려처리 확인 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재공통 (APC) | 일괄 반려 (ARJ) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APC.ARJ** | **기능번호** | **AP.APC.4** |
| **기능설명** | 결재관련 항목에서 여러건을 일괄적으로 반려 처리하는 기능 | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재관련 승인대기 상태의 건에 대해 일괄적으로 반려처리하는 기능 **(사후결재는 기능 미사용)**  일반/보안/메일 결재관리 리스트에서 승인가능한 상태를 확인하여 반려를 처리  <일반/보안 결재>  \* Step 1: 결재할 대상이 결재가 가능한 상태인지 확인하여 승인할 목록을 선택  승인가능상태 확인 : 승인대기 / 승인가능Flag / 전송상태 / 사후결재전송여부  \* Step 2: 전문 $APPROVE\_BATCH를 통해 승인처리  Tag PROCID: 승인(1)/**반려(2**)  Tag Reason : 반려사유  Tag ApproveKind : **사전결재(0)**/사후결재(1)  Tag APPROVESEQS : 승인요청 정보Key들 (delemiter는 0x02, 최대 100개 제한)  Tag APPROVER : 대결재자정보 안남김(0)/대결재자정보 안남김(1)  Tag APPROVEUSERKIND : 일반결재(0)/보안결재(1)/PCURL결재(2)  \* Step 3: 전문 $APPROVE\_BATCH\_ACK 서버로 부터 받아 정상 반려처리 확인  <메일 결재>  \* Step 1: 결재할 대상이 결재가 가능한 상태인지 확인하여 승인할 목록을 선택  승인가능상태 확인 : 승인대기 / 승인가능Flag / 전송상태 / 사후결재전송여부  \* Step 2: 전문 $EMAIL\_APPROVE\_BATCH를 통해 반려처리  Tag PROCID: 승인(1)/**반려(2)**  Tag Reason : 반려사유  Tag EmailApproveSeqS: 메일전송 요청Key들 (delemiter는 0x02, 최대 100개 제한)  \* Step 3: 전문 $EMAIL\_APPROVE\_BATCH \_ACK 서버로 부터 받아 정상 반려처리 확인 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재상세보기 공통  (APDC) | 파일미리보기 (FPV) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APDC.FPV** | **기능번호** | **AP.APDC.1** |
| **기능설명** | 결재자가 결재를 하기 위해 결재전에 파일을 확인하는 기능 | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재자가 결재를 하기 위해 결재전에 파일을 확인하는 기능  반출은 업무망에서 반입은 인터넷망에서만 보기가 가능  \* Step 1: 결재할 대상이 파일들이 미리보기가 가능한 상태인지 확인  승인가능상태 확인 : 반입반출에 대한 결재자 접속망 / 승인대기 / 전송상태  **사후승인건 추가 확인 : 전송날짜기준으로 사후결재파일 보관주기(AFTERAPPROVEFILE)가 남아 있는지**  **보관주기가 지났으면 원본저장(BACKUPSAVECYCLE) 주기로 파일이 남아 있는지 확인**  \* Step 2: 미리보기를 할 파일을 선택하여 전문 $FILEDOWNLOAD64를 통해 파일요청  Tag TransSeq : 전송순번  Tag FileName : 파일명  Tag FileKey : FileKey  \* Step 3: 전문 $FILEDOWNLOAD64\_ACK 서버로 부터 받아 임시 파일에 저장  \* Step 4: 임시파일에 저장된 파일을 실행 (Shell명령어로 OS의 실행프로그램이 자동연결)  \* Step 5: 임시파일에 저장된 파일 설정시간 후 제거   * 참고 : 32bit 파일전송을 사용하는 구 NetLink 1.x에서는 전문 $FILEDOWNLOAD로 사용 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 결재상세보기 공통  (APDC) | 승인 후 파일 미리보기 (AFFP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.APDC.AFFP** | **기능번호** | **AP.APDC.2** |
| **기능설명** | 결재자가 승인 후에도 파일을 확인하는 기능 | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재자가 승인 후에도 파일을 확인하는 기능 (업무망에서만 가능)  원본저장 기능을 서버에서 사용 중이여야 하며, 원본저장 주기 내에서만 가능  \* Step 1: 결재한 대상이 파일들이 보기가 가능한 상태인지 확인  승인가능상태 확인 : 결재자 업무망 접속 / 승인상태 / 전송상태 /  서버의 원본 저장 기능 확인 / 원본저장(BACKUPSAVECYCLE) 주기  \* Step 2: 미리보기를 할 파일을 선택하여 전문 $FILEDOWNLOAD64를 통해 파일요청  Tag TransSeq : 전송순번  Tag FileName : 파일명  Tag FileKey : FileKey  \* Step 3: 전문 $FILEDOWNLOAD64\_ACK 서버로 부터 받아 임시 파일에 저장  \* Step 4: 임시파일에 저장된 파일을 실행 (Shell명령어로 OS의 실행프로그램이 자동연결)  \* Step 5: 임시파일에 저장된 파일 설정시간 후 제거   * 참고 : 32bit 파일전송을 사용하는 구 NetLink 1.x에서는 전문 $FILEDOWNLOAD로 사용 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 파일결재 상세보기 연동 (FADV) | 파일미리보기 특정프로그램 연동 (FPP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.FADV.FPP** | **기능번호** | **AP.FADV.1** |
| **기능설명** | 이미지 파일의 경우 그림판으로만 파일 미리보기 연동. (옵션) | | |
| **입력** |  | | |
| **출력** |  | | |
| 미리보기 기능 중 실행 프로그램 연동  이미지 파일의 경우 그림판 실행 (옵션)  \* Step 1~3: 기능 코드 AP.APDC.FPV와 동일  \* Step 4: 임시파일에 저장된 파일이 이미지 파일이면 그림판으로 실행  \* Step 5: 임시파일에 저장된 파일 설정시간 후 제거 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 메일결재 상세보기  (MADV) | 본문보기 (MMV) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.MADV.MMV** | **기능번호** | **AP.MADV.1** |
| **기능설명** | 메일의 본문내용을 파일로 다운받아 확인한다. | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재자가 메일 결재를 하기 위해 결재전에 메일본문을 확인하는 기능  반출은 업무망에서 반입은 인터넷망에서만 보기가 가능  \* Step 1: 결재할 대상의 메일본문이 미리보기가 가능한 상태인지 확인  승인가능상태 확인 : 반입반출에 대한 결재자 접속망 / 승인대기 / 전송상태  \* Step 2: 전문 $EMAIL\_FILEDOWNLOAD64를 통해 메일본문 파일 요청  Tag EmailSeq : 전송순번  Tag FileName : 파일명  Tag FileKey : FileKey  \* Step 3: 전문 $EMAIL\_FILEDOWNLOAD64\_ACK 서버로 부터 받아 임시 파일에 저장  \* Step 4: 임시파일에 저장된 파일을 실행 (Shell명령어로 OS의 실행프로그램이 자동연결)  \* Step 5: 임시파일에 저장된 파일 설정시간 후 제거 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 결재(AP) | 메일결재 상세보기  (MADV) | 첨부파일 미리보기 (MFV) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **AP.MADV.MFV** | **기능번호** | **AP.MADV.2** |
| **기능설명** | 메일의 본문내용을 파일로 다운받아 확인한다. | | |
| **입력** |  | | |
| **출력** |  | | |
| 결재자가 메일 결재를 하기 위해 결재전에 첨부파일을 확인하는 기능  반출은 업무망에서 반입은 인터넷망에서만 보기가 가능  \* Step 1: 결재할 대상의 첨부파일이 미리보기가 가능한 상태인지 확인  승인가능상태 확인 : 반입반출에 대한 결재자 접속망 / 승인대기 / 전송상태  \* Step 2: 미리보기를 할 파일을 선택하여 전문 $EMAIL\_FILEDOWNLOAD64를 통해 파일요청  Tag TransSeq : 전송순번  Tag FileName : 파일명  Tag FileKey : FileKey  \* Step 3: 전문 $EMAIL\_FILEDOWNLOAD64\_ACK 서버로 부터 받아 임시 파일에 저장  \* Step 4: 임시파일에 저장된 파일을 실행 (Shell명령어로 OS의 실행프로그램이 자동연결)  \* Step 5: 임시파일에 저장된 파일 설정시간 후 제거 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| PCURL 사용신청 (PU) | 결재라인 (APL) | 고정 결재라인 (SAL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **PU.APL.SAL** | **기능번호** | **PU.APL.1** |
| **기능설명** | 서버에서 받은 결재 Line 대로 결재를 받는다. | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| PCURL 사용신청 (PU) | 결재라인 (APL) | 결재자 선택 (AS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **PU.APL.AS** | **기능번호** | **PU.APL.2** |
| **기능설명** | 검색(이름 / 부서), 조직도상에서 선택, | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| PCURL 사용신청 (PU) | 요청사유 (RRD) | 사용신청 사유 (URR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **PU.RRD.URR** | **기능번호** | **PU.RRD.1** |
| **기능설명** | 사용 신청한 URL 에 대한 사유 입력 | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| PCURL 사용신청 (PU) | 정책적용범위 (PC) | 승인된 URL 사용 범위 (AUC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **PU.PC.AUC** | **기능번호** | **PU.PC.1** |
| **기능설명** | 개인 또는 부서 선택을 통해 사용 승인된 URL 의 사용범위를 지정. | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| PCURL 사용신청 (PU) | 신청가능 URL 검색  (UUS) | URL 또는 그룹명 검색 (UGS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **PU.UUS.UGS** | **기능번호** | **PU.UUS.1** |
| **기능설명** | 신청 가능 URL 전체 목록에서 URL 명 또는 그룹명으로 검색. | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| PCURL 사용신청 (PU) | 신청가능 URL 추가  (UUA) | 사용 신청할 URL 추가 (UUA) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **PU.UUA.UUA** | **기능번호** | **PU.UUA.1** |
| **기능설명** | 신청 가능 URL 목록에서 그룹별로 URL 사용 신청. | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| PCURL 사용신청 (PU) | 신청가능 URL 삭제  (UUD) | 사용 신청할 URL 삭제 (UUD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **PU.UUD.UUD** | **기능번호** | **PU.UUD.1** |
| **기능설명** | 신청할 URL 목록에서 그룹별로 URL 사용 신청 삭제 | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| PCURL 사용신청 (PU) | 신규 URL 사용 신청 (NUR) | 신규 URL 사용 신청 (NUR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **PU.NUR.NUR** | **기능번호** | **PU.NUR.1** |
| **기능설명** | 신청 가능 URL 목록 외에 신규로 신청할 신규 URL 신청. | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 업데이트(UP) | 기본 (BASE) | 로그인 (LO) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UP.BASE.LO** | **기능번호** | **UP.BASE.1** |
| **기능설명** | 로그인 시점에 자동으로 패치 진행 | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: 로그인(전문 $BIND) 후 응답(전문 $BIND\_ACK)으로 서버에 업데이트가 있는지 확인  Tag ClientUpgrade 값이 1이면 업데이트 있음  \* Step 2: 서버의 업데이트 버전 정보와 현재 Agent의 버전 정보를 비교하여 서버버전이 높으면 업데이트.  Tag ClientVersion : 업데이트 버전정보  \* Step 3: 업데이트 프로세스 실행( 예) NLUpgrade.exe )과 Agent 종료  \* Step 4: 업데이트 프로세스는 서버주소, 포트, TLS버전 정보를 전달받아 접속을 함  \* Step 5: 전문 $CLIENTUPGRADE으로 업데이트 파일을 요청하여 $CLIENTUPGRADE\_ACK로 수신함  \* Step 6: 전문 $CLIENTUPGRADE\_ACK의 업데이트 정보를 확인 및 서버와 통신 종료  Tag CliVersion : 업데이트 버전  Tag SIGN : 전자서명정보  \* Step 7: 압축을 해제 및 파일이동 프로세스( 예)HMove.exe )를 실행하고 NLUpgrade.exe는 종료  \* Step 8: 파일이동 프로세스( 예)HMove.exe )는 파일을 설치 디렉토리에 이동하고 Agent 실행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 업데이트(UP) | 노티 (NT) | 서버 노티 (SNT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UP.NT.SNT** | **기능번호** | **UP.NT.1** |
| **기능설명** | 업데이트 파일이 존재하면 Agent 사용 중 서버 알림 노티를 수신받고 패치 진행. | | |
| **입력** |  | | |
| **출력** |  | | |
| 사용자가 Agent 사용 중에 서버로부터 업데이트 노티를 받아 업데이트를 진행  \* Step 1: 서버로부터 노티 전문 $CLIENTVERSION를 수신하여 Tag CliVersion과 현재 버전을 비교하여 사용자에게 업데이트 있음을 표시  \* Step 2: 업데이트 프로세스 실행( 예) NLUpgrade.exe )과 Agent 종료  \* Step 3: 업데이트 프로세스는 서버주소, 포트, TLS버전 정보를 전달받아 접속을 함  \* Step 4: 전문 $CLIENTUPGRADE으로 업데이트 파일을 요청하여 $CLIENTUPGRADE\_ACK로 수신함  \* Step 5: 전문 $CLIENTUPGRADE\_ACK의 업데이트 정보를 확인 및 서버와 통신 종료  Tag CliVersion : 업데이트 버전  Tag SIGN : 전자서명정보  \* Step 6: 압축을 해제 및 파일이동 프로세스( 예)HMove.exe )를 실행하고 NLUpgrade.exe는 종료  \* Step 7: 파일이동 프로세스( 예)HMove.exe )는 파일을 설치 디렉토리에 이동하고 Agent 실행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 업데이트(UP) | 백그라운드 (BG) | 백그라운드 패치파일 수신 (BG) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UP.BG.BG** | **기능번호** | **UP.BG.1** |
| **기능설명** | 업데이트 파일이 존재하면 Agent 사용 중 서버 알림 노티를 수신받고 백그라운드로 파일 수신. 파일 수신 완료 후 알림 메시지를 나타내거나 Agent 재 실행 시 로그인 전 수신받은 패치파일이 존재할 경우 사용자에게 알림. | | |
| **입력** |  | | |
| **출력** |  | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 업데이트(UP) | 메뉴 (MENU) | 업그레이드 메뉴 (MENU) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UP.BG.BG** | **기능번호** | **UP.BG.1** |
| **기능설명** | 사용자가 업그레이드 메뉴를 사용하여 패치 진행. | | |
| **입력** |  | | |
| **출력** |  | | |
| 사용자가 메뉴를 통해 직접 업데이트를 확인하고 업데이트 하는 기능  \* Step 1: 사용자가 메뉴의 업데이트 확인을 실행  \* Step 2: 전문 $UserInfoCheck로 사용자의 최신정책 정보를 요청  \* Step 3: 전문 $UserInfoCheck\_ack로 사용자의 최신정책 중 업데이트 사항이 있는지 확인  Tag ClientUpgrade 값이 1이면 업데이트 있음  \* Step 4: 서버의 업데이트 버전 정보와 현재 Agent의 버전 정보를 비교하여 서버버전이 높으면 업데이트.  Tag ClientVersion : 업데이트 버전정보  \* Step 5: 업데이트 프로세스 실행( 예) NLUpgrade.exe )과 Agent 종료  \* Step 6: 업데이트 프로세스는 서버주소, 포트, TLS버전 정보를 전달받아 접속을 함  \* Step 7: 전문 $CLIENTUPGRADE으로 업데이트 파일을 요청하여 $CLIENTUPGRADE\_ACK로 수신함  \* Step 8: 전문 $CLIENTUPGRADE\_ACK의 업데이트 정보를 확인 및 서버와 통신 종료  Tag CliVersion : 업데이트 버전  Tag SIGN : 전자서명정보  \* Step 9: 압축을 해제 및 파일이동 프로세스( 예)HMove.exe )를 실행하고 NLUpgrade.exe는 종료  \* Step 10: 파일이동 프로세스( 예)HMove.exe )는 파일을 설치 디렉토리에 이동하고 Agent 실행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 업데이트(UP) | 부서별 패치 (DTP) | 부서별 기본 패치 (DTP) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UP.DTP.DTP** | **기능번호** | **UP.DTP.1** |
| **기능설명** | 부서별 설정에 따라 필요한 부서에만 서버 노티 수신 후 기본 패치 진행 | | |
| **입력** |  | | |
| **출력** |  | | |
| 서버 설정(PATCHMODE)에 따라 설정된 부서 및 사용자만 업데이트를 진행  업데이트에 포함되는 사용자는 기능코드 UP.NT.SNT 와 동일하게 노티 방식으로 진행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 업데이트(UP) | 부서별 패치 (DTP) | 부서별 백그라운드 패치 (DTPB) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UP.DTP.DTPB** | **기능번호** | **UP.DTP.2** |
| **기능설명** | 부서별 설정에 따라 필요한 부서에만 서버 노티 수신 후 백그라운드 패치 진행. | | |
| **입력** |  | | |
| **출력** |  | | |
| 서버 설정(PATCHMODE)에 따라 설정된 부서 및 사용자만 업데이트를 진행  업데이트에 포함되는 사용자는 기능코드 UP.BG.BG 와 동일하게 노티 방식으로 진행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 업데이트(UP) | 업데이트 내역 (UPD) | 업데이트 내역 공지 (UPD) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UP.UPD.UPD** | **기능번호** | **UP.UPD.1** |
| **기능설명** | 업데이트 완료 후 사용자에게 업데이트 내역을 알림. | | |
| **입력** |  | | |
| **출력** |  | | |
| 업데이트 후 사이트 요청 및 커스텀마이징에 따라 업데이트 내역을 달리 표시한다.  Case 1 : 특정 웹페이지 호출  Case 2 : 업데이트 이력 TXT 파일 실행  Case 3 : 공지사항 기능 사용 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| URL 리다이렉션 (UR) | List (LIST) | White List (WL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UR.LIST.WL** | **기능번호** | **UR.LIST.1** |
| **기능설명** | 서버에 설정된 White List 정보를 수신받아 URL.xml 파일에 기록. | | |
| **입력** |  | | |
| **출력** |  | | |
| URL리다이렉션 기능은 인터넷을 사용하지 못하는 업무망에서 인터넷망에 페이지를 띄워 주는 기능  업무망의 URL을 인터넷망에 띄워 줄 URL List를 서버로부터 받아와서 관리  \* Step 1: 로그인 후 전문 $URLLIST을 통해 서버로부터 URL List를 받아옴  \* Step 2: 받아온 URL List가 White/Black 방식인지를 확인하여 XML파일을 생성  \* Step 3: Iexplorer 등록된 BHO가 이벤트가 발생한 URL의 도메인이 XML파일의 URL list에 있는지 확인  \* Step 4: BHO는 URL을 Agent에 전달  \* Step 5: 업무망 Agent는 전달 받은 URL을 전문 $SubDataExchange 로 서버에 요청  \* Step 6: 인터넷망 Agent는 서버로부터 전문 $URLAUTOTRANS 를 수신하여 URL확인  \* Step 7: 기본 브라우저에 수신한 URL을 전달   * 참고 : URL은 \*를 지원함 예) \*.naver.com | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| URL 리다이렉션 (UR) | List (LIST) | Black List (BL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UR.LIST.BL** | **기능번호** | **UR.LIST.2** |
| **기능설명** | 서버에 설정된 Black List 정보를 수신받아 URL.xml 파일에 기록. | | |
| **입력** |  | | |
| **출력** |  | | |
| URL리다이렉션 기능은 인터넷을 사용하지 못하는 업무망에서 인터넷망에 페이지를 띄워 주는 기능  업무망의 URL을 인터넷망에 띄워 줄 URL List를 서버로부터 받아와서 관리  \* Step 1: 로그인 후 전문 $URLLIST을 통해 서버로부터 URL List를 받아옴  \* Step 2: 받아온 URL List가 White/Black 방식인지를 확인하여 XML파일을 생성  \* Step 3: Iexplorer 등록된 BHO가 이벤트가 발생한 URL의 도메인이 XML파일의 URL list에 없는지 확인  \* Step 4: BHO는 URL을 Agent에 전달  \* Step 5: 업무망 Agent는 전달 받은 URL을 전문 $SubDataExchange 로 서버에 요청  \* Step 6: 인터넷망 Agent는 서버로부터 전문 $URLAUTOTRANS 를 수신하여 URL확인  \* Step 7: 기본 브라우저에 수신한 URL을 전달   * 참고 : URL은 \*를 지원함 예) \*.naver.com | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| URL 리다이렉션 (UR) | List (LIST) | Exception List (EL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UR.LIST.EL** | **기능번호** | **UR.LIST.3** |
| **기능설명** | 서버에 설정된 예외 List 정보를 수신받아 URL.xml 파일에 기록 | | |
| **입력** |  | | |
| **출력** |  | | |
| 기능코드 UR.LIST.WL, UR.LIST.BL의 예외적으로 연동하는 기능  \* Step 1: 로그인 후 전문 $URLLIST을 통해 서버로부터 URL List를 받아옴  \* Step 2: 받아온 URL List가 White/Black 방식인지를 확인하여 XML파일을 생성  \* Step 3: 받아온 URL List 중 예외 URL이 있을 경우 XML파일에 추가 예외 URL등록  \* Step 4: Iexplorer 등록된 BHO가 이벤트가 발생한 URL이 XML파일의 예외URL list에 없는지 확인  \* Step 5: URL 예외URL list에 있는 경우 설정이 White면 Black방식으로 연결을 하지 않고  설정이 Black이면 White방식으로 BHO는 URL을 Agent에 전달  \* Step 6: 업무망 Agent는 전달 받은 URL을 전문 $SubDataExchange 로 서버에 요청  \* Step 7: 인터넷망 Agent는 서버로부터 전문 $URLAUTOTRANS 를 수신하여 URL확인  \* Step 8: 기본 브라우저에 수신한 URL을 전달   * 참고 : 예외 URL은 \*을 지원하지 않음 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| URL 리다이렉션 (UR) | 자동전환방식 (TYPE) | 포워딩 (FW) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UR.TYPE.FW** | **기능번호** | **UR.TYPE.1** |
| **기능설명** | URL 리다이렉션된 페이지 변경 (옵션) | | |
| **입력** |  | | |
| **출력** |  | | |
| URL자동 전송된 페이지를 옵션에 따라 브라우저를 컨트롤 하는 옵션기능  (참고 기능코드 UR.LIST.WL, UR.LIST.BL, UR.LIST.EL)  BHO가 확인한 URL을 설정된 안내페이지로 변경 (파일이나 안내URL) | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| URL 리다이렉션 (UR) | 자동전환방식 (TYPE) | 유지 (NO) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UR.TYPE.NO** | **기능번호** | **UR.TYPE.2** |
| **기능설명** | URL 리다이렉션된 페이지 유지 (옵션) | | |
| **입력** |  | | |
| **출력** |  | | |
| URL자동 전송된 페이지를 옵션에 따라 브라우저를 컨트롤 하는 옵션기능  (참고 기능코드 UR.LIST.WL, UR.LIST.BL, UR.LIST.EL)  BHO가 확인한 URL을 변경없이 연동만 진행 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| URL 리다이렉션 (UR) | 자동전환방식 (TYPE) | 닫기 (CL) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UR.TYPE.CL** | **기능번호** | **UR.TYPE.3** |
| **기능설명** | URL 리다이렉션된 페이지 닫음 (옵션) | | |
| **입력** |  | | |
| **출력** |  | | |
| URL자동 전송된 페이지를 옵션에 따라 브라우저를 컨트롤 하는 옵션기능  (참고 기능코드 UR.LIST.WL, UR.LIST.BL, UR.LIST.EL)  BHO가 확인한 URL의 브라우저를 닫음 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| URL 리다이렉션 (UR) | 설정(CF) | 사용유무 설정 (US) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UR.CF.US** | **기능번호** | **UR.CF.1** |
| **기능설명** | 환경설정을 통해 사용자가 사용 유무를 설정. | | |
| **입력** |  | | |
| **출력** |  | | |
| 서버설정에 따라 URL 리다이렉션 기능을 제공하며,  URL 리다이렉션 기능을 제공 할 때 사용자가 기능 유무를 설정  \* Step 1: 로그인시 전문 $BIND\_ACK 의 Tag UrlRedirection 값을 기능 제공 여부 확인  Tag UrlRedirection : 0: 미사용 1: 업무망->인터넷망 2: 인터넷망->업무망  \* Step 2: 기능 제공이된 경우 사용자 저장 값을 읽어 사용여부를 ON/OFF함 (기본 ON) | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| URL 리다이렉션 (UR) | Chrome(CE) | Chorme 환경에서 동작 (CE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UR.CE.CE** | **기능번호** | **UR.CE.1** |
| **기능설명** | IE 가 아닌 Chrome 환경에서 URL 리다이렉션 동작. | | |
| **입력** |  | | |
| **출력** |  | | |
| 기능코드 UR.LIST.WL, UR.LIST.BL, UR.LIST.EL와 달리 Chrome BHO를 등록하지 않고  Chrome Browser의 이벤트를 후킹하는 방식  \* Step 1: 로그인 후 전문 $URLLIST을 통해 서버로부터 URL List를 받아옴  \* Step 2: 받아온 URL List가 White/Black 방식인지를 확인하여 map에 보관  \* Step 3: Chrome 주소창의 이벤트 후킹  \* Step 4: 후킹된 이벤트가 연결관련이면 URL list에 있는지 확인하여 기능 코드 기능코드 UR.LIST.WL, UR.LIST.BL, UR.LIST.EL 과 같이 동작  \* Step 5: URL을 전문 $SubDataExchange 로 서버에 요청  \* Step 6: 인터넷망 Agent는 서버로부터 전문 $URLAUTOTRANS 를 수신하여 URL확인  \* Step 7: 기본 브라우저에 수신한 URL을 전달   * 참고 : Chrome은 기능 정의서 UR.TYPE.NO는 지원하지 않음 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 전송(TR) | Text 기본 (OT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.OT.OT** | **기능번호** | **CT.TR.1** |
| **기능설명** | 클립보드의 DATA 중 Text 정보만 전송한다. | | |
| **입력** | 사용자 입력(단축키 or 클립보드 툴바) | | |
| **출력** | 전송 성공 여부 | | |
| \* Step 1: OS 상의 클립보드 데이터 접근을 위해 Open 한다.(OpenClipboard)  \* Step 2: OS 에서 Text 형태의 클립보드 데이터를 가져온다.(GetClipboardData)  Cf) Text 형태의 클립보드 데이터 타입 : CF\_TEXT, CF\_OEMTEXT, CF\_UNICODETEXT  \* Step 3: 클립보드 데이터 전송 전 클립보드 데이터 용량 또는 횟수가 유효한지 여부를 검사하고  전송 가능 여부를 판별한다.  \* Step 3: Text 형태의 클립보드 데이터를 프로토콜 CLIPBOARDTXT 를 통해 서버로 전송한다.  \* Step 4: 클립보드 데이터 서버 전송 후 결과를 사용자에게 알려준다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 전송(TR) | Text 파일 변환 (FT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.TR.FT** | **기능번호** | **CT.TR.2** |
| **기능설명** | 클립보드의 DATA 중 Text 정보를 txt 파일 형태로 변환하여 파일로 전송한다. | | |
| **입력** | 사용자 입력(단축키 or 클립보드 툴바) | | |
| **출력** | 전송 성공 여부 | | |
| \* Step 1: OS 상의 클립보드 데이터 접근을 위해 Open 한다.(OpenClipboard)  \* Step 2: OS 에서 Text 형태의 클립보드 데이터를 가져와 txt 파일 형태로 변환한다.  (GetClipboardData)  Cf) Text 형태의 클립보드 데이터 타입 : CF\_TEXT, CF\_OEMTEXT, CF\_UNICODETEXT  \* Step 3: 클립보드 데이터 전송 전 클립보드 데이터 용량 또는 횟수가 유효한지 여부를 검사하고  전송 가능 여부를 판별한다.  \* Step 3: 클립보드 텍스트 Data를 변환한 txt 파일을 프로토콜 TR\_FILEUPLOAD, TRANSREQ 를 통해  서버로 전송한다.  \* Step 4: 변환된 txt 파일 서버 전송 후 결과를 사용자에게 알려준다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 전송(TR) | 이미지 기본 (OI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.TR.OI** | **기능번호** | **CT.TR.3** |
| **기능설명** | 클립보드의 DATA 중 이미지 정보를 bmp 형태로 변환하여 전송한다. | | |
| **입력** | 사용자 입력(단축키 or 클립보드 툴바) | | |
| **출력** | 전송 성공 여부 | | |
| \* Step 1: OS 상의 클립보드 데이터 접근을 위해 Open 한다.(OpenClipboard)  \* Step 2: OS 에서 이미지 형태의 클립보드 데이터를 가져온다.(GetClipboardData)  Cf) 이미지 형태의 클립보드 데이터 타입 : CF\_BITMAP, CF\_DIB  \* Step 3: 클립보드 데이터 전송 전 클립보드 데이터 용량 또는 횟수가 유효한지 여부를 검사하고  전송 가능 여부를 판별한다.  \* Step 3: 이미지 형태의 클립보드 데이터를 BMP 타입으로 변환 후 프로토콜 CLIPBOARDTXT 를 통해  서버로 전송한다.  \* Step 4: 클립보드 데이터 서버 전송 후 결과를 사용자에게 알려준다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 전송(TR) | 이미지 파일 변환 (FI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.TR.FI** | **기능번호** | **CT.TR.4** |
| **기능설명** | 클립보드의 DATA 중 이미지 정보를 bmp 형태로 변환하여 파일로 전송한다. | | |
| **입력** | 사용자 입력(단축키 or 클립보드 툴바) | | |
| **출력** | 전송 성공 여부 | | |
| \* Step 1: OS 상의 클립보드 데이터 접근을 위해 Open 한다.(OpenClipboard)  \* Step 2: OS 에서 이미지 형태의 클립보드 데이터를 가져와 bmp 파일 형태로 변환한다.  (GetClipboardData)  Cf) Text 형태의 클립보드 데이터 타입 : CF\_BITMAP, CF\_DIB  \* Step 3: 클립보드 데이터 전송 전 클립보드 데이터 용량 또는 횟수가 유효한지 여부를 검사하고  전송 가능 여부를 판별한다.  \* Step 3: 클립보드 이미지 Data를 변환한 bmp 파일을 프로토콜 TR\_FILEUPLOAD, TRANSREQ 를 통해  서버로 전송한다.  \* Step 4: 변환된 bmp 파일 서버 전송 후 결과를 사용자에게 알려준다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 전송(TR) | Object (OBTY) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.TR.OBTY** | **기능번호** | **CT.TR.5** |
| **기능설명** | 클립보드의 DATA 를 압축하여 전송한다. Html 형태의 데이터 전송 시 이미지 경로 정보를 수정한다. | | |
| **입력** | 사용자 입력(단축키 or 클립보드 툴바) | | |
| **출력** | 전송 성공 여부 | | |
| \* Step 1: OS 상의 클립보드 데이터 접근을 위해 Open 한다.(OpenClipboard)  \* Step 2: OS 에서 얻어지는 클립보드 데이터를 타입별로 dat 임시파일로 저장한다.  (GetClipboardData)  클립보드 데이터 타입이 CF\_METAFILEPICT 인 경우 OS 상의 클립보드 메모리 이미지 경로를  확인하여 이미지 데이터를 함께 dat 파일에 삽입한다.  \* Step 3: 저장된 dat 형식의 임시파일들을 하나의 hsz 파일 형태로 압축한다.  HTML 형태의 데이터인 경우 hsz 파일로 압축하기 전 HTML 의 이미지 경로를 수정한다.  \* Step 3: 클립보드 데이터 전송 전 클립보드 데이터 용량 또는 횟수가 유효한지 여부를 검사하고  전송 가능 여부를 판별한다.  \* Step 3: 압축된 hsz 파일을 프로토콜 CLIPBOARDOBJECT 를 통해 서버로 전송한다.  \* Step 4: 압축된 hsz 파일 서버 전송 후 결과를 사용자에게 알려준다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 수신(RV) | Text 기본(OT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.RV.OT** | **기능번호** | **CT.RV.1** |
| **기능설명** | 클립보드 Text 정보를 수신받고 OS에 Text 정보를 Set 한다. | | |
| **입력** | 수신된 클립보드 데이터 | | |
| **출력** | OS 상에 Set 한 클립보드 데이터(Text) | | |
| \* Step 1: 서버로부터 CLIPBOARDTXT 노티를 수신 받는다.  \* Step 2: 서버로부터 수신받은 클립보드 데이터를 파싱하여 임시폴더에 dat 파일 형태로 저장한다.  Cf) 임시 파일명 : cur\_clip.dat  \* Step 3: 클립보드 데이터 수신이 완료되면 OS 상의 클립보드 데이터 접근을 위해 Open 한다.  (OpenClipboard)  \* Step 4: CLIPBOARDTXT 수신 프로토콜의 DataType 을 확인하고 dat 파일의 텍스트 정보를 읽어  클립보드 데이터타입 CF\_UNICODETEXT 으로 OS에 Set 한다.(SetClipboardData) | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 수신(RV) | Text 파일 변환(FT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.RV.FT** | **기능번호** | **CT.RV.2** |
| **기능설명** | Txt 파일을 수신 받은 후 Txt 파일의 Text 정보를 OS에 Set 한다. | | |
| **입력** | 수신된 클립보드 데이터 | | |
| **출력** | OS 상에 Set 한 클립보드 데이터(Text) | | |
| \* Step 1: 서버로부터 AUTODOWNLOAD 노티를 통해 hsz 형태의 압축 파일을 수신받는다.  \* Step 2: 수신된 hsz 형태의 파일을 hsz 헤더의 type 정보를 확인하여 임시폴더에 txt 형태의 파일로  압축을 해제한다.  Cf) 텍스트 type : 1, 이미지 type : 2  \* Step 3: 압축 해제가 완료되면 OS 상의 클립보드 데이터 접근을 위해 Open 한다.  (OpenClipboard)  \* Step 4: 압축 해제된 txt 파일을 로드하여 txt 파일의 Text 정보를 읽어들인다.  \* Step 5: txt 파일의 Text 정보를 클립보드 데이터타입 CF\_UNICODETEXT 으로 OS에 Set 한다.  (SetClipboardData) | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 수신(RV) | 이미지 기본(OI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.RV.OI** | **기능번호** | **CT.RV.3** |
| **기능설명** | 클립보드 이미지 정보를 수신받고 OS에 이미지 정보를 Set 한다. | | |
| **입력** | 수신된 클립보드 데이터 | | |
| **출력** | OS 상에 Set 한 클립보드 데이터(이미지) | | |
| \* Step 1: 서버로부터 CLIPBOARDTXT 노티를 수신 받는다.  \* Step 2: 서버로부터 수신받은 클립보드 데이터를 파싱하여 임시폴더에 dat 파일 형태로 저장한다.  Cf) 임시 파일명 : cur\_clip.dat  \* Step 3: 클립보드 데이터 수신이 완료되면 OS 상의 클립보드 데이터 접근을 위해 Open 한다.  (OpenClipboard)  \* Step 4: CLIPBOARDTXT 수신 프로토콜의 DataType을 확인하고 dat파일을 HBITMAP 메모리에 로드한다.  \* Step 5: HBITMAP 메모리 데이터를 클립보드 데이터타입 CF\_BITMAP 으로 OS에 Set 한다.  (SetClipboardData) | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 수신(RV) | 이미지 파일 변환(FI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.RV.FI** | **기능번호** | **CT.RV.4** |
| **기능설명** | Bmp 파일을 수신 받은 후 Bmp 파일의 이미지 정보를 OS 에 Set 한다. | | |
| **입력** | 수신된 클립보드 데이터 | | |
| **출력** | OS 상에 Set 한 클립보드 데이터(이미지) | | |
| \* Step 1: 서버로부터 AUTODOWNLOAD 노티를 통해 hsz 형태의 압축 파일을 수신받는다.  \* Step 2: 수신된 hsz 형태의 파일을 hsz 헤더의 type 정보를 확인하여 임시폴더에 bmp 형태의 파일로  압축을 해제한다.  Cf) 텍스트 type : 1, 이미지 type : 2  \* Step 3: 압축 해제가 완료되면 OS 상의 클립보드 데이터 접근을 위해 Open 한다.  (OpenClipboard)  \* Step 4: 압축 해제된 bmp 파일을 HBITMAP 메모리에 로드한다..  \* Step 5: HBITMAP 메모리 데이터를 클립보드 데이터타입 CF\_BITMAP 으로 OS에 Set 한다.  (SetClipboardData) | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 수신(RV) | Obejct(OBTY) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.RV.OBTY** | **기능번호** | **CT.RV.5** |
| **기능설명** | 수신받은 클립보드 Data 를 압축 해제하여 OS 에 Set 한다. | | |
| **입력** | 수신된 클립보드 데이터 | | |
| **출력** | OS 상에 Set 한 클립보드 데이터(Object) | | |
| \* Step 1: 서버로부터 CLIPBOARDOBJECT 노티를 통해 hsz 형태의 압축 파일을 수신받는다.  \* Step 2: 수신된 hsz 형태의 파일을 임시폴더에 dat 형태의 파일로 압축을 해제한다.  \* Step 3: 압축 해제가 완료되면 OS 상의 클립보드 데이터 접근을 위해 Open 한다.  (OpenClipboard)  \* Step 4: 압축 해제된 dat 파일의 클립보드 데이터 타입을 확인하여 타입정보와 함께 dat 파일을  OS 에 Set 한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 수신 알림형식(RAT) | 팝업(PU) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.RAT.PU** | **기능번호** | **CT.RAT.1** |
| **기능설명** | 클립보드 수신 시 수신 완료 팝업 메시지를 표출한다. | | |
| **입력** | 클립보드 수신 완료 노티. | | |
| **출력** | 수신완료 메시지 팝업창. | | |
| # 서버로부터 CLIPBOARDOTXT , CLIPBOARDOBJECT , AUTODOWNLOAD 노티를 통해  hsz 형태의 압축 파일 수신 및 압축해제후 클립보드 데이터가 OS 상에 Set 이 완료되면  사용자에게 클립보드 수신 완료 메시지 팝업 창을 표출한다.  Cf) 클립보드 데이터 수신 노티   1. 기본(Text or 이미지) : CLIPBOARDTXT 2. 파일변환(Text or 이미지) : AUTODOWNLOAD 3. Obejct : CLIPBOARDOBEJCT   \* Step 1: 위에 설명된 클립보드 수신 및 OS 상에 클립보드 데이터 Set 이 완료되면  사용자 개인설정 정보인 SecureGate.ini 에 저장된 CLIPTYPE 값을 확인한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK  \* Step 2: CLIPTYPE 값이 0 이면 사용자에게 메시지 팝업창으로 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 수신 알림형식(RAT) | 트레이(TRAY) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.RAT.TRAY** | **기능번호** | **CT.RAT.2** |
| **기능설명** | 클립보드 수신 시 수신 완료 트레이를 표출한다. | | |
| **입력** | 클립보드 수신 완료 노티. | | |
| **출력** | 수신완료 트레이. | | |
| # 서버로부터 CLIPBOARDOTXT , CLIPBOARDOBJECT , AUTODOWNLOAD 노티를 통해  hsz 형태의 압축 파일 수신 및 압축해제 후 클립보드 데이터가 OS 상에 Set 이 완료되면  사용자에게 클립보드 수신 완료 메시지 팝업 창을 표출한다.  Cf) 클립보드 데이터 수신 노티   1. 기본(Text or 이미지) : CLIPBOARDTXT 2. 파일변환(Text or 이미지) : AUTODOWNLOAD 3. Obejct : CLIPBOARDOBEJCT   \* Step 1: 위에 설명된 클립보드 수신 및 OS 상에 클립보드 데이터 Set 이 완료되면  사용자 개인설정 정보인 SecureGate.ini 에 저장된 CLIPTYPE 값을 확인한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK  \* Step 2: CLIPTYPE 값이 1 이면 사용자에게 트레이로 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 수신 알림형식(RAT) | 팝업 & 트레이(PT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.RAT.PT** | **기능번호** | **CT.RAT.3** |
| **기능설명** | 클립보드 수신 시 수신 완료 팝업메시지 창과 트레이를 동시에 표출한다. | | |
| **입력** | 클립보드 수신 완료 노티. | | |
| **출력** | 수신완료 메시지 팝업창 과 트레이. | | |
| # 서버로부터 CLIPBOARDOTXT , CLIPBOARDOBJECT , AUTODOWNLOAD 노티를 통해  hsz 형태의 압축 파일 수신 및 압축해제 후 클립보드 데이터가 OS 상에 Set 이 완료되면  사용자에게 클립보드 수신 완료 메시지 팝업 창을 표출한다.  Cf) 클립보드 데이터 수신 노티   1. 기본(Text or 이미지) : CLIPBOARDTXT 2. 파일변환(Text or 이미지) : AUTODOWNLOAD 3. Obejct : CLIPBOARDOBEJCT   \* Step 1: 위에 설명된 클립보드 수신 및 OS 상에 클립보드 데이터 Set 이 완료되면  사용자 개인설정 정보인 SecureGate.ini 에 저장된 CLIPTYPE 값을 확인한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK  \* Step 2: CLIPTYPE 값이 2 이면 사용자에게 메시지 팝업창과 트레이를 동시에 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 수신 알림형식(RAT) | 설정변경(SV) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.RAT.SV** | **기능번호** | **CT.RAT.4** |
| **기능설명** | 환경설정에서 클립보드 수신 알림형식을 변경한다. | | |
| **입력** | 사용자 설정 입력. | | |
| **출력** | 설정값 변경. | | |
| \* Step 1: 사용자에게 클립보드 수신 완료 표출 방법은 환경설정에서 설정한다.  \* Step 2: 환경설정에서 클립보드 수신 완료 표출 방법을 팝업으로 설정하면  사용자 개인설정 정보가 담긴 SecureGate.ini 에 CLIPTYPE 값을 0으로 변경한다.  Cf) ClipType 값 - 팝업 : 0 , 트레이 : 1 , 팝업 & 트레이 : 2  \* Step 3: 위에 설명된 클립보드 수신 및 OS 상에 클립보드 데이터 Set 이 완료되면  사용자 개인설정 정보인 SecureGate.ini 에 CLIPTYPE 값을 변경한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 단축키(KB) | 전송(TR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.KB.TR** | **기능번호** | **CT.KB.1** |
| **기능설명** | Agent 실행 중 단축키를 입력하면 클립보드 DATA 를 전송한다. | | |
| **입력** | 사용자 단축키 입력 | | |
| **출력** | 클립보드 전송 | | |
| \* Step 1: Agent 로그인 완료 후 사용자 개인설정 정보가 저장된 SecureGate.ini 파일의  CLIPBOARD 태그에 저장된 단축키 설정 정보를 로드한다.  SecureGate.ini 파일에 단축키 정보가 존재하지 않으면 단축키 설정 정보를 로드하기 전SGConf.xml 에 설정된 값을 SecureGate.ini 파일에 단축키 설정 정보로 저장한다.  \* Step 2: 로드된 단축키 정보를 통해 OS 상에 HotKey 등록을 한다. (RegisterHotKey)  \* Step 3: WM\_HOTKEY 메시지 핸들러를 통해 사용자 단축키 입력에 대해 모니터링한다.  \* Step 4: WM\_HOTKEY 메시지 핸들러를 통해 사용자 단축키 입력이 들어오면 클립보드 전송을 수행한다.  Cf) 클립보드 전송은 본 문서 기능번호 CT.TR.1 ~ CT.TR.5 참조. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 단축키(KB) | 복사 후 전송(CNTR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.KB.CNTR** | **기능번호** | **CT.KB.2** |
| **기능설명** | Agent 실행 중 단축키를 입력하면 클립보드 전송이 자동으로 수행된다. | | |
| **입력** | 사용자 단축키 입력 | | |
| **출력** | 클립보드 전송 | | |
| \* Step 1: Agent 로그인 완료 후 사용자 개인설정 정보가 저장된 SecureGate.ini 파일의  CLIPBOARD 태그에 저장된 단축키 설정 정보를 로드한다.  SecureGate.ini 파일에 단축키 정보가 존재하지 않으면 단축키 설정 정보를 로드하기 전SGConf.xml 에 설정된 값을 SecureGate.ini 파일에 단축키 설정 정보로 저장한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK  \* Step 2: 로드된 단축키 정보를 통해 OS 상에 HotKey 등록을 한다. (RegisterHotKey)  \* Step 3: WM\_HOTKEY 메시지 핸들러를 통해 사용자 단축키 입력에 대해 모니터링한다.  \* Step 4: WM\_HOTKEY 메시지 핸들러를 통해 사용자 단축키 입력이 들어오면  보내려는 타겟(클립보드 데이터)을 OS 상에 Set 한다.  \* Step 5: 전송하려는 클립보드 데이터가 OS 상에 Set 이 완료되면 클립보드 전송을 수행한다.  Cf) 클립보드 전송은 본 문서 기능번호 CT.TR.1 ~ CT.TR.5 참조. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 클립보드(CT) | 단축키(KB) | 설정(SV) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **CT.KB.SV** | **기능번호** | **CT.KB.3** |
| **기능설명** | 환경설정에서 단축키 설정을 변경한다. | | |
| **입력** | 사용자 단축키 선택. | | |
| **출력** | 변경된 단축키 Reset | | |
| \* Step 1: 사용자는 클립보드 단축키 설정 변경을 환경설정을 통해 수행한다.  \* Step 2: 환경설정 팝업 창을 열고 사용자가 원하는 키조합을 선택한다.  Cf) 키조합 : Window Key, Ctrl Key, Alt Key, Shift Key, (숫자 | 영문 알파벳 | F1~F12) 조합이 가능.  \* Step 3: 사용자에 의해 키조합 선택이 완료되고 환경설정 팝업 창의 확인 버튼을 누르면  변경된 단축키 정보를 사용자 개인설정정보가 저장된 SecureGate.ini 파일 CLIPBOARD 태그  값으로 저장한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK  \* Step 4: 변경된 단축키 정보 저장이 완료되면 OS 상의 HotKey 정보를 변경된 단축키로 Reset 한다.  (ResetHotKey) | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **대분류** | **중분류** | | **소분류** | | |
| 클립보드(CT) | 마우스우클릭(MRC) | | 클립보드 자동전송(MRC) | | |
| **기능코드** | | **CT.MRC.MRC** | | **기능번호** | **CT.MRC.1** |
| **기능설명** | | 마우스 우클릭 메뉴를 사용하여 클립보드 Data를 자동으로 전송한다. | | | |
| **입력** | | 마우스 우클릭 선택. | | | |
| **출력** | | 클립보드 데이터 전송. | | | |
| \* Step 1: 망연계 Agent 설치 시 마우스 우클릭 dll (COM) 을 OS 에 등록한다.  Cf) COM 등록은 regsvr32.exe 를 통해 등록.  \* Step 2: 설치를 통해 등록이 완료되면 마우스 우클릭 메뉴에서 클립보드 전송 메뉴가 표출된다.  \* Step 3: 클립보드 전송 메뉴를 선택하면 Agent로 특정 파라미터 “C”를 전송한다.  \* Step 4: 특정 인자값 C를 전달받은 Agent 는 OS 상에 미리 저장된 클립보드 데이터를 가져온다.  \* Step 5: OS 상의 클립보드 데이터를 로드하는데 성공하면 클립보드 전송을 수행한다. | | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 화면잠금(UL) | 설정(CF) | 사용 유무 설정(USE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UL.CF.USE** | **기능번호** | **UL.CF.1** |
| **기능설명** | 환경설정에서 화면잠금 사용 유무를 사용자가 설정한다. | | |
| **입력** | 사용자 화면잠금 사용유무 선택. | | |
| **출력** | 화면잠금 실행/미실행. | | |
| \* Step 1: 사용자는 화면잠금 사용 유무 설정 변경을 환경설정을 통해 수행한다.  \* Step 2: 환경설정 팝업 창을 열고 화면잠금 체크박스를 체크 또는 해제한다.  \* Step 3: 사용자가 환경설정 팝업 창의 확인 버튼을 누르면 화면잠금 체크박스 체크 유무에 따라  개인설정정보가 저장된 SecureGate.ini 파일 SCRLOCKUSE 태그 값으로 “Y or N” 을 저장한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK  \* Step 4: Agent 는 SecureGate.ini 파일의 SCRLOCKUSE 태그 값을 로드하여 설정된 정보에 맞게  화면잠금 동작 여부를 Reset 한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 화면잠금(UL) | 설정(CF) | 시간 설정(TIME) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UL.CF.TIME** | **기능번호** | **UL.CF.2** |
| **기능설명** | 환경설정에서 서버에 설정된 시간을 넘지 않는 범위에서 화면잠금 시간을 설정한다. | | |
| **입력** | 사용자 화면잠금 시간 입력 | | |
| **출력** | 설정된 시간 후 화면잠금 실행 | | |
| \* Step 1: 사용자는 화면잠금 시간 설정 변경을 환경설정을 통해 수행한다.  \* Step 2: 환경설정 팝업 창을 열고 화면잠금 시간 입력창에 원하는 시간값을 입력한다.  Cf) 사용자가 변경하기를 원하는 화면 잠금 시간 값은 서버 환경변수 SCRLOCKTIME 값을  초과 할수 없으며 0 으로 설정하게 되면 화면잠금은 동작하지 않는다.  입력값의 단위는 분.  \* Step 3: 사용자가 환경설정 팝업 창의 확인 버튼을 누르면 변경한 화면잠금 시간 값을  개인설정정보가 저장된 SecureGate.ini 파일 SCRLOCKVAL 태그 값으로 저장한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK  \* Step 4: Agent 는 SecureGate.ini 파일의 SCRLOCKVAL 태그 값을 로드하여 설정된 정보에 맞게  화면잠금 동작 시간을 Reset 한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 화면잠금(UL) | 동작(WO) | 화면잠금 실행(LOCK) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UL.WO.LOCK** | **기능번호** | **UL.WO.1** |
| **기능설명** | Agent가 지정된 시간 동안 사용이 안되면 자동적으로 화면잠금 동작을 수행한다. | | |
| **입력** | 화면잠금 사용유무와 화면잠금 시간 | | |
| **출력** | 화면잠금 해제 팝업 창 | | |
| \* Step 1: Agent 는 로그인 후 화면잠금 사용유무와 화면잠금 시간 값을 확인한다.  화면잠금 사용유무와 화면잠금 시간 값은 개인설정정보가 저장된 SecureGate.ini 파일  SCRLOCKUSE 값과 SCRLOCKTIME 값을 로드하여 확인한다.  \* Step 2: 설정되어 있는 시간 만큼 Agent에 사용자 입력이 없으면 메인 화면을 숨기고  화면잠금 해제 팝업 창을 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 화면잠금(UL) | 동작(WO) | 화면잠금 해제(UNLO) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UL.WO.UNLO** | **기능번호** | **UL.WO.2** |
| **기능설명** | 사용자 Password 정보를 입력하여 화면잠금을 해제하고 Agent 메인화면을 표출한다. | | |
| **입력** | 사용자 Password | | |
| **출력** | Agent 메인 화면 | | |
| \* Step 1: Agent 가 화면잠금 동작을 수행중이라면 화면잠금 해제 팝업 창이 표출되어 있다.  \* Step 2: 사용자가 화면잠금 해제를 수행하려면 화면잠금 해제 팝업 창의 비밀번호 입력란에  사용자 자신의 Password를 입력 한 후 화면잠금 해제 버튼을 누른다.  \* Step 3: 사용자가 화면잠금 해제 버튼을 누르면 프로토콜 “CLIENTUNLOCK” 을 통해  서버로 사용자 Password가 올바른지 여부를 확인한다.  \* Step 4: 서버로부터 “CLIENTUNLOCK” Ack 를 수신 받고 Password 가 올바른지 여부를  확인한 다음 Password 가 정확하면 화면잠금 해제를 수행하고  Password 가 틀리면 사용자에게 화면잠금 실패 안내 메시지를 표출한다.  \* Step 5: Agent 가 화면잠금 해제가 되면 화면잠금 해제 팝업창은 사라지고  Agent 메인 화면이 표출된다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| 화면잠금(UL) | 동작(WO) | 프로그램 종료(EXIT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **UL.WO.EXIT** | **기능번호** | **UL.WO.3** |
| **기능설명** | Agent가 지정된 시간 동안 사용이 안되면 자동적으로 종료한다. | | |
| **입력** | Agent 종료 시간 | | |
| **출력** | Agent 종료 | | |
| \* Step 1: Agent는 로그인 후 서버로부터 환경변수 SCRLOCKTIME 값을 수신받는다.  \* Step 2: Agent는 서버로부터 수신 받은 SCRLOCKTIME 값을 확인하고  자동종료 사용 유무와 자동종료 시간을 확인한다.  Cf) 자동종료 사용 유무는 개인설정이 아니며 서버환경변수 SCRLOCKTIME 값에 의해 동작한다.  SCRLOCKTIME 값이 0이면 사용안함. 1 이상이면 설정된 시간 후 Agent 자동종료 수행.  해당 기능은 화면잠금 기능과 같은 환경변수를 사용함으로써  화면잠금 기능과 동시 수행이 불가함.  \* Step 3: 자동종료 사용 일 경우 설정되어 있는 시간 만큼 Agent에 사용자 입력이 없으면  Agent 는 자동종료된다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 패스워드 관리(PM) | 초기 패스워드 변경(IPC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.PM.IPC** | **기능번호** | **ET.PM.1** |
| **기능설명** | ID/PW 로그인시 초기패스워드로 인증할 경우 자동으로 패스워드 변경 창을 팝업한다. | | |
| **입력** | 초기 패스워드 | | |
| **출력** | 패스워드 변경 | | |
| \* Step 1: 사용자는 처음 로그인 수행 시 초기비밀번호 로그인을 수행한다.  \* Step 2: 로그인 수행 시 사용자 인증이 완료되면 Agent는 초기 비밀번호로 로그인을  수행하였는지 여부를 검사한다.  \* Step 3: 초기 비밀번호로 로그인을 수행한 것을 확인하면 사용자에게 비밀번호 변경이 필요하다는  메시지 팝업창을 표출하고 사용자 비밀번호 변경창을 팝업한다.  \* Step 4: 비밀번호 변경창에서 사용자가 변경할 비밀번호 입력 후 비밀번호변경 버튼을 클릭한다.  \* Step 5: 비밀번호변경 버튼을 클릭하면 Agent는 변경할 비밀번호의 유효성을 검사한다.  변경할 비밀번호가 유효하지 않을 경우 사용자에게 안내 메시지 팝업창을 표출한다.  Cf) 유효성 : 비밀번호 길이, 특수문자 입력, 복잡도 검사, 연속된 문자…  \* Step 6: 변경할 비밀번호의 유효성 검사가 통과되면 프로토콜 CHANGEPASSWD 를 통해  서버로 해당 사용자의 비밀번호 변경을 요청한다.  \* Step 7: 사용자 비밀번호 변경이 완료되면 비밀번호 변경완료 메시지 팝업창을 표출하고  동시에 Agent 메인 화면을 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 패스워드 관리(PM) | 패스워드 변경(PC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.PM.PC** | **기능번호** | **ET.PM.2** |
| **기능설명** | 지정된 기간이 초과되면 자동으로 패스워드 변경 창을 팝업한다. | | |
| **입력** | 사용자 패스워드 변경 만기일. | | |
| **출력** | 패스워드 변경. | | |
| \* Step 1: 사용자는 처음 로그인 수행 시 서버로부터 패스워드 변경 유무를 수신받는다.  패스워드 변경 만기일 설정은 서버환경변수 PASSWDEXPIREDDAYS의 값으로 설정하고  Bind\_ack 를 통해 패스워드 변경 유무를 수신받는다.  Cf) bind\_ack 태그 값 PASSWORDEXPIRED 사용 - 0: 미사용, 1: Notify, 2:Popup (강제변경)  \* Step 2: 비밀번호 변경이 필요함을 확인하면 자동으로 사용자 비밀번호 변경창을 팝업한다.  \* Step 3: 비밀번호 변경창에서 사용자가 변경할 비밀번호 입력 후 비밀번호변경 버튼을 클릭한다.  \* Step 4: 비밀번호변경 버튼을 클릭하면 Agent는 변경할 비밀번호의 유효성을 검사한다.  변경할 비밀번호가 유효하지 않을 경우 사용자에게 안내 메시지 팝업창을 표출한다.  Cf) 유효성 : 비밀번호 길이, 특수문자 입력, 복잡도 검사, 연속된 문자…  \* Step 5: 변경할 비밀번호의 유효성 검사가 통과되면 프로토콜 CHANGEPASSWD 를 통해  서버로 해당 사용자의 비밀번호 변경을 요청한다.  \* Step 6: 사용자 비밀번호 변경이 완료되면 비밀번호 변경완료 메시지 팝업창을 표출하고  동시에 Agent 메인 화면을 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 패스워드 관리(PM) | 패스워드 변경 History(PCH) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.PM.PCH** | **기능번호** | **ET.PM.3** |
| **기능설명** | 패스워드 변경 시 이전 패스워드 History를 검사한다. | | |
| **입력** | 변경하려는 사용자 패스워드 | | |
| **출력** | 이전 패스워드 사용 유무 | | |
| \* Step 1: 사용자가 비밀번호 변경 수행 시 이전에 사용되었던 비밀번호인지 여부를 검사하기 위해  서버로 DB 조회를 수행한다.  \* Step 2: 서버 DB 조회 후 이전에 사용된 비밀번호일 경우 사용자 안내 메시지 팝업 후  사용자에게 다시 비밀번호 설정할 것을 안내한다.  \* Step 3. 서버 DB 조회 후 이전에 사용된 비밀번호가 아니라면 프로토콜 CHANGEPASSWD 를 통해  비밀번호 변경을 수행하고 Agent 메인화면이 표출된다.  Cf) 이전 비밀번호 사용 유무 검사 DB 조회 쿼리   |  | | --- | | "\  select 'POSSIBLE' tag, cast( not exists( \  select \* \  from ( \  select p.seq, p.user\_seq, p.before\_pw user\_pass, p.pw\_upt\_dttm \  from tbL\_user\_passwd\_his p \  , tbl\_user\_info u \  where p.user\_seq=u.user\_seq \  and u.user\_seq=%s \  order by p.pw\_upt\_dttm desc \  limit 2 \  ) a \  where a.user\_pass='%s' \  ) as integer) val" | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 패스워드 관리(PM) | 변경 방식(PCT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.PM.PCT** | **기능번호** | **ET.PM.4** |
| **기능설명** | 서버 설정에 따라 강제 변경 또는 2주후 변경이 가능하게 한다. | | |
| **입력** | 패스워드 변경방식 | | |
| **출력** | 패스워드 강제변경 또는 2주후 변경. | | |
| \* Step 1: 사용자는 처음 로그인 수행 시 패스워드 만기일 만료 유무와 변경방식을 확인 하기 위해  서버로부터 PASSWORDEXPIRED 값을 수신 받는다.  PASSWORDEXPIRED 는 bind\_ack 수행 시 태그 값으로 전달된다.  Cf) PASSWORDEXPIRED - 0: 미사용, 1: Notify, 2:Popup (강제변경)  \* Step 2: 비밀번호 변경이 필요함을 확인하면 자동으로 사용자 비밀번호 변경창을 팝업한다.  PASSWORDEXPIRED 값이 2인 경우 비밀번호 변경창 팝업 시 강제변경이 되도록  “2주간 표시안함” , “다음에 변경” 버튼이 숨겨지고 그 외의 값일 경우에는  2개의 버튼이 표출된다.  \* Step 3: 강제변경 사용 시에는 비밀번호 변경을 해야만 Agent 메인화면이 표출되고  변경하지 않을 시에는 그대로 종료된다..  강제변경 사용을 안 할 시에는 “다음에 변경”, “2주간 표시안함” 버튼이 표출되고  “다음에 변경” 버튼을 클릭하면 Agent는 메인화면이 표출되고  다음 로그인 시 비밀번호 변경 창이 팝업된다.  “2주후 표시안함” 버튼을 클릭하면 Agent는 메인화면이 표출되고  2주후 비밀번호 변경 창이 팝업된다.  Cf) 서버 환경변수 설정방법  PASSWDEXPIREDMETHOD 값이 1이면 사용자 비밀번호 강제변경 지원.  PASSWDEXPIREDMETHOD 값이 0이면 사용자 비밀번호 강제변경 지원안함. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 공인인증서(CT) | NPKI 공인인증서 전송(NPSE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CT.NPSE** | **기능번호** | **ET.CT.1** |
| **기능설명** | NPKI 공인인증서 파일을 반대편망으로 전송한다.  내/외부 Agent 가 모두 실행되어야 가능하다. | | |
| **입력** | NPKI 공인인증서 파일 | | |
| **출력** | NPKI 공인인증서 파일 전송. | | |
| \* Step 1: Agent는 NPKI 공인인증서 전송 시작 시 반대편망 Agent 접속을 확인하라는  안내 메시지 팝업을 표출한다.  Cf) NPKI 공인인증서 수신은 실시간 방식으로 수행되므로 반대편망 Agent 접속이  되어 있어야 수신이 가능하다.  \* Step 2: 사용자 안내메시지 팝업 후 사용자는 어느 경로에 있는 NPKI 인증서를 전송할 것인지  선택한다.  Cf) 하드디스크 or 이동식디스크 중 선택.  - 하드디스크 : OS가 설치된 디스크 경로.  - 이동식디스크 : USB와 같은 이동식 디스크 경로  \* Step 3: Step2 에서 사용자가 선택한 경로에 따라 지정된 경로의 NPKI 인증서 모든 파일들을  탐색한다.  Cf) 일반적인 NPKI 인증서 파일 경로  - OS 가 설치된 하드디스크 : %USERPROFILE%\AppData\LocalLow\NPKI,  OS 설치 드라이브\Program Files(x86)\NPKI  - 이동식디스크 : 선택된 드라이브의 루트 경로. 루트\NPKI  \* Step 4: Step3 에서 탐색된 NPKI 인증서 모든 파일을 임시 폴더에 SG.pcf 파일로 변환한다.  \* Step 5: 변환된 SG.pcf 파일을 프로토콜 CLIPBOARDTXT 를 통해 클립보드 방식으로 전송한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 공인인증서(CT) | NPKI 공인인증서 선택 전송(NPSSE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CT.NPSSE** | **기능번호** | **ET.CT.2** |
| **기능설명** | NPKI 공인인증서 파일을 사용자가 직접 선택하여 반대편망으로 전송한다.  내/외부 Agent 가 모두 실행되어야 가능하다. | | |
| **입력** | 선택된 NPKI 공인인증서 파일 | | |
| **출력** | NPKI 공인인증서 파일 전송. | | |
| \* Step 1: Agent는 NPKI 공인인증서 전송 시작 시 반대편망 Agent 접속을 확인하라는  안내 메시지 팝업을 표출한다.  Cf) NPKI 공인인증서 수신은 실시간 방식으로 수행되므로 반대편망 Agent 접속이  되어 있어야 수신이 가능하다.  \* Step 2: 사용자 안내메시지 팝업 후 사용자는 NPKI 공인인증서 선택 창을 통해 전송하려는  공인인증서 파일을 선택한다.  Cf) NPKI 공인인증서 선택 창은 하드디스크 또는 이동식디스크를 선택이 가능하며  해당 경로의 NPKI 인증서 목록을 리스트 창에 로드하여 표출한다.  - 하드디스크 : OS가 설치된 디스크 경로.  - 이동식디스크 : USB와 같은 이동식 디스크 경로  \* Step 3: Step2 에서 사용자가 선택한 NPKI 인증서 파일을 임시 폴더에 SG.pcf 파일로 변환한다.  \* Step 4: 변환된 SG.pcf 파일을 프로토콜 CLIPBOARDTXT 를 통해 클립보드 방식으로 전송한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 공인인증서(CT) | NPKI 공인인증서 수신(NPRE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CT.NPRE** | **기능번호** | **ET.CT.3** |
| **기능설명** | NPKI 공인인증서 파일을 수신받고 지정된 경로에 복사한다. | | |
| **입력** | NPKI 공인인증서 파일 수신 노티. | | |
| **출력** | NPKI 공인인증서 파일 수신. | | |
| \* Step 1: 서버로부터 CLIPBOARDTXT 노티를 수신 받는다.  \* Step 2: 서버로부터 수신받은 데이터를 파싱하여 임시폴더에 dat 파일 형태로 저장한다.  Cf) 임시 파일명 : cur\_clip.dat  \* Step 3: NPKI 공인인증서 데이터 수신이 완료되면 type 을 검사하여 공인인증서 파일인지 여부를  검사한다.  Cf) type : 1 - 클립보드 Text 데이터  2 – 클립보드 bmp 데이터  3 – 공인인증서 데이터.  \* Step 4: 공인인증서 데이터임이 확인되면 NPKI 지정된 경로에 인증서 파일을 해제한다.  Cf) 일반적인 NPKI 인증서 파일 경로  - OS가 설치된 하드디스크 : %USERPROFILE%\AppData\LocalLow\NPKI,  OS 설치 드라이브\Program Files(x86)\NPKI  - 이동식디스크 : 선택된 드라이브의 루트 경로. 루트\NPKI | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 공인인증서 전송(CT) | GPKI 공인인증서 전송(GPSE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CT.GPSE** | **기능번호** | **ET.CT.4** |
| **기능설명** | GPKI 공인인증서 파일을 반대편망으로 전송한다.  내/외부 Agent 가 모두 실행되어야 가능하다. | | |
| **입력** | GPKI 공인인증서 파일 | | |
| **출력** | GPKI 공인인증서 파일 전송. | | |
| \* Step 1: Agent는 GPKI 공인인증서 전송 시작 시 반대편망 Agent 접속을 확인하라는  안내 메시지 팝업을 표출한다.  Cf) GPKI 공인인증서 수신은 실시간 방식으로 수행되므로 반대편망 Agent 접속이  되어 있어야 수신이 가능하다.  \* Step 2: 사용자 안내메시지 팝업 후 사용자는 어느 경로에 있는 GPKI 인증서를 전송할 것인지  선택한다.  Cf) 하드디스크 or 이동식디스크 중 선택.  - 하드디스크 : OS가 설치된 디스크 경로.  - 이동식디스크 : USB와 같은 이동식 디스크 경로  \* Step 3: Step2 에서 사용자가 선택한 경로에 따라 지정된 경로의 GPKI 인증서 모든 파일들을  탐색한다.  Cf) 일반적인 NPKI 인증서 파일 경로  - OS가 설치된 하드디스크 : OS설치 드라이브\GPKI  - 이동식디스크 : 선택된 드라이브의 루트 경로. 루트\GPKI  \* Step 4: Step3 에서 탐색된 GPKI 인증서 모든 파일을 임시 폴더에 SG.pcf 파일로 변환한다.  \* Step 5: 변환된 SG.pcf 파일을 프로토콜 CLIPBOARDTXT 를 통해 클립보드 방식으로 전송한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 공인인증서(CT) | GPKI 공인인증서 수신(GPRE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CT.GPRE** | **기능번호** | **ET.CT.5** |
| **기능설명** | GPKI 공인인증서 파일을 수신받고 지정된 경로에 복사한다. | | |
| **입력** | GPKI 공인인증서 파일 수신 노티 | | |
| **출력** | GPKI 공인인증서 파일 수신. | | |
| \* Step 1: 서버로부터 CLIPBOARDTXT 노티를 수신 받는다.  \* Step 2: 서버로부터 수신받은 데이터를 파싱하여 임시폴더에 dat 파일 형태로 저장한다.  Cf) 임시 파일명 : cur\_clip.dat  \* Step 3: GPKI 공인인증서 데이터 수신이 완료되면 type 을 검사하여 공인인증서 파일인지 여부를  검사한다.  Cf) type : 1 - 클립보드 Text 데이터  2 – 클립보드 bmp 데이터  3 – 공인인증서 데이터.  \* Step 4: 공인인증서 데이터임이 확인되면 GPKI 지정된 경로에 인증서 파일을 해제한다.  Cf) 일반적인 GPKI 인증서 파일 경로  - OS가 설치된 하드디스크 : OS설치 드라이브\GPKI | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 공지사항(NOTE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.NOTE** | **기능번호** | **ET.AN.1** |
| **기능설명** | 로그인 후 서버로부터 공지사항 정보를 수신받아 사용자에게 알림 팝업창을 표출한다. | | |
| **입력** | 공지사항 노티 | | |
| **출력** | 공지사항 팝업 창 | | |
| \* Step 1: 사용자가 공지사항 표출 대상자인 경우 서버로부터 BOARDNOTIFY 노티를 수신 받는다.  \* Step 2: 공지받은 노티를 수신받은 Agent 는 공지사항 내용 조회를 위해  프로토콜 “HSCKQUERY” 를 통해 서버로 DB조회 쿼리를 전송한다.  \* Step 3: 서버로 DB 조회가 완료되면 응답값으로 공지사항 내용을 수신받아 사용자에게  공지사항 정보를 팝업한다.  Cf) 공지사항 DB 조회 쿼리.   |  | | --- | | CREATE OR REPLACE FUNCTION public.func\_nl\_boardnotify(  i\_userid character varying,  i\_date character varying)  RETURNS TABLE(o\_idx bigint, o\_titile character varying, o\_content character varying, o\_url character varying, o\_stdate character varying, o\_enddate character varying, o\_deptseq bigint, o\_deptname character varying, o\_read character varying)  LANGUAGE 'plpgsql'  COST 100  VOLATILE  ROWS 1000  AS $BODY$  DECLARE  SQL VARCHAR; -- 쿼리  BEGIN  RETURN QUERY  SELECT A.IDX, A.TITLE  -- , CAST(REPLACE(A.CONTENTS, chr(13)||chr(10), '%13%10') AS  , CAST(REGEXP\_REPLACE(CONTENTS, '(\n|\r)\n', '%13%10', 'g') AS  VARCHAR) CONTENTS  , CAST('NOT EXIST URL' AS VARCHAR) URL  , CAST(A.DTSTART AS VARCHAR) DTSTART, CAST(A.DTEND AS VARCHAR) DTEND, A.DEPT\_SEQ, A.DEPT\_NAME  , CAST(CASE WHEN U.READ IS NULL OR U.READ ='' THEN '0' ELSE U.READ END AS VARCHAR) RF  FROM (  SELECT B.\*, BD.DEPT\_SEQ, D.DEPT\_NAME, BD.USER\_SEQ  FROM TBL\_BOARD\_INFO B  , (  SELECT D.IDX, U.DEPT\_SEQ, U.USER\_SEQ  FROM TBL\_BOARD\_DEPT D  , TBL\_USER\_INFO U  WHERE D.TYPE='0'  AND U.USER\_ID=i\_userid  UNION ALL  SELECT D.IDX, D.DEPT\_SEQ, U.USER\_SEQ  FROM TBL\_BOARD\_DEPT D  , TBL\_USER\_INFO U  WHERE D.TYPE!='0'  AND U.USER\_ID=i\_userid  AND U.DEPT\_SEQ=D.DEPT\_SEQ  ) BD  , TBL\_DEPT\_INFO D  WHERE B.IDX=BD.IDX  AND BD.DEPT\_SEQ=D.DEPT\_SEQ  AND B.USEYN='1' -- 사용여부 확인  AND B.LOC IN ('0', '2') -- 전체 및 NetLink 공지사항망  ) A  LEFT OUTER JOIN TBL\_BOARD\_USER U ON A.IDX=U.IDX AND A.USER\_SEQ=U.USER\_SEQ  WHERE (U.NOMORE IS NULL OR U.NOMORE !='1')  AND ( CASE WHEN i\_date IS NULL THEN A.DTINSERT > ''  WHEN i\_date='' THEN A.DTINSERT > ''  ELSE A.DTINSERT >= i\_date END)  ;  END;  $BODY$;  ALTER FUNCTION public.func\_nl\_boardnotify(character varying, character varying)  OWNER TO hsck; | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 온/오프라인(ON/OFF) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.ONOFF** | **기능번호** | **ET.AN.2** |
| **기능설명** | 로그인 완료 또는 네트워크 끊김현상 발생 시 사용자에게 온/오프라인 정보를 트레이로 표출한다. | | |
| **입력** | 네트워크 상태 확인 | | |
| **출력** | 온/오프라인 트레이 | | |
| \* Step 1: Agent는 네트워크 상태에 따라 온/오프라인을 트레이로 표출한다..  \* Step 2: 네트워크 상태가 Off 되거나 Link Check 실패 발생 시 사용자에게 오프라인 트레이를  표출한다.  \* Step 3: Agent 시작 시 로그인이 성공하거나 네트워크 상태가 Off 라인에서 On 되었을 경우  Agent 는 Reconnect 를 시도하고 성공하면 온라인 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 파일전송 결재 승인대기(FTAS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.FTAS** | **기능번호** | **ET.AN.3** |
| **기능설명** | 파일전송 건에 대해 결재가 필요한 건수가 존재하면 서버로부터 노티를 수신받아 해당 결재자에게 트레이로 표출한다. | | |
| **입력** | 승인대기 알림노티 | | |
| **출력** | 승인대기 알림 트레이 | | |
| \* Step 1: 사용자가 결재자인 경우 승인대기 건수가 존재하면 서버로부터 APPROVECOUNT 노티를  수신 받는다.  \* Step 2: APPROVECOUNT 노티를 수신받은 Agent 는 사용자(결재자)가 인식할 수 있도록  화면 오른쪽 하단에 파일전송 승인대기 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 이메일 결재 승인대기(EMAS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.EMAS** | **기능번호** | **ET.AN.4** |
| **기능설명** | 이메일에 대해 결재가 필요한 건수가 존재하면 서버로부터 노티를 수신받아 해당 결재자에게 트레이로 표출한다. | | |
| **입력** | 이메일 승인대기 알림노티 | | |
| **출력** | 이메일 승인대기 알림 트레이 | | |
| \* Step 1: 사용자가 이메일 결재자인 경우 이메일 승인대기 건수가 존재하면 서버로부터 EMAIL\_APPROVE\_NOTIFY 노티를 수신 받는다.  \* Step 2: EMAIL\_APPROVE\_NOTIFY 노티를 수신받은 Agent는 데이터를 파싱하여  이메일 결재가 필요한 남은 건수를 확인한다.  Cf) 결재 남은 건수 : EMAIL\_APPROVE\_NOTIFY 노티의 EMAILAPPROVECOUNT 값을 확인.  \* Step 3: Agent 는 사용자(결재자)가 인식할 수 있도록 남은 결재 건수 정보를 화면 오른쪽 하단에  이메일 승인대기 알림 트레이로 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 클립보드 결재 승인대기(CTAS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.CTAS** | **기능번호** | **ET.AN.5** |
| **기능설명** | 클립보드 파일 전송에 대해 결재가 필요한 건수가 존재하면 서버로부터 노티를 수신받아 해당 결재자에게 트레이로 표출한다. | | |
| **입력** | 클립보드 승인대기 알림노티 | | |
| **출력** | 클립보드 승인대기 알림 트레이 | | |
| \* Step 1: 사용자가 결재자인 경우 클립보드 파일 전송 관련 승인대기 건수가 존재하면  서버로부터 APPROVECOUNT 노티를 수신 받는다.  \* Step 2: APPROVECOUNT 노티를 수신받은 Agent 는 데이터를 파싱하여 승인대기 알림 종류를  확인한다.  Cf ) 승인대기 종류 알림 : APPROVECOUNT 노티의 APPROVEUSERKIND 값을 확인  APPROVEUSERKIND 값이 0 이면 일반결재.  APPROVEUSERKIND 값이 1 이면 보안결재  APPROVEUSERKIND 값이 2 이면 PCURL 결재  APPROVEUSERKIND 값이 3 이면 클립보드 결재.  \* Step 3: APPROVECOUNT 노티를 수신받은 Agent 는 사용자(결재자)가 인식할 수 있도록  화면 오른쪽 하단에 클립보드 승인대기 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 정보보안 결재 승인대기(SEAS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.SEAS** | **기능번호** | **ET.AN.6** |
| **기능설명** | 보안결재를 사용할 경우 보안결재자에게 서버로부터 노티를 수신받아 트레이로 표출한다. | | |
| **입력** | 보안결재 승인대기 알림노티 | | |
| **출력** | 보안결재 승인대기 알림 트레이 | | |
| \* Step 1: 사용자가 보안결재자인 경우 보안결재 승인대기 건수가 존재하면 서버로부터 PRIVACY\_APPROVE\_NOTIFY 노티를 수신 받는다.  \* Step 2: PRIVACY\_APPROVE\_NOTIFY 노티를 수신받은 Agent는 데이터를 파싱하여  보안 결재가 필요한 남은 건수를 확인한다.  Cf) 보안 결재 남은 건수 : PRIVACY\_APPROVE\_NOTIFY 노티의 PRIVACYAPPRPOVECOUNT  값을 확인.  \* Step 3: Agent 는 사용자(보안결재자)가 인식할 수 있도록 남은 보안 결재 건수 정보를  화면 오른쪽 하단에 보안결재 승인대기 알림 트레이로 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | PCURL 결재 승인대기(PUAS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.PUAS** | **기능번호** | **ET.AN.7** |
| **기능설명** | PCURL 사용 신청에 대해 결재가 필요할 경우 서버로부터 노티를 수신받아 해당 결재자에게 트레이로 표출한다. | | |
| **입력** | PCURL 승인대기 알림노티 | | |
| **출력** | PCURL 승인대기 알림노티 트레이 | | |
| \* Step 1: 사용자가 결재자인 경우 PCURL 신청에 대한 승인대기 건수가 존재하면  서버로부터 APPROVECOUNT 노티를 수신 받는다.  \* Step 2: APPROVECOUNT 노티를 수신받은 Agent 는 데이터를 파싱하여 승인대기 알림 종류를  확인한다.  Cf ) 승인대기 종류 알림 : APPROVECOUNT 노티의 APPROVEUSERKIND 값을 확인  APPROVEUSERKIND 값이 0 이면 일반결재.  APPROVEUSERKIND 값이 1 이면 보안결재  APPROVEUSERKIND 값이 2 이면 PCURL 결재  APPROVEUSERKIND 값이 3 이면 클립보드 결재.  \* Step 3: APPROVECOUNT 노티를 수신받은 Agent 는 사용자(결재자)가 인식할 수 있도록  화면 오른쪽 하단에 PCURL 승인대기 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 파일전송 결재 승인완료(FTAC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.FTAC** | **기능번호** | **ET.AN.8** |
| **기능설명** | 파일 전송 건에 대해 서버로부터 승인 완료 노티를 수신받아 해당 사용자(요청자)에게 트레이로 표출한다. | | |
| **입력** | 승인 완료 노티 | | |
| **출력** | 승인 완료 알림노티 트레이 | | |
| \* Step 1: 사용자가 파일전송을 수행 한 후 결재자가 결재(승인)를 수행하면  서버로부터 APPROVEACTION\_NOTIFY 노티를 수신 받는다.  \* Step 2: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 데이터를 파싱하고  ACTION 값을 확인하여 결재종류를 확인한 다음 APPROVEUSERKIND 값을 확인하여  파일전송 결재인지 여부도 확인한다.  Cf) Action 값 : 1(승인), 2(반려),3(등록완료),4(등록취소) -> 3,4번은 PCURL 에서만 사용.  ApproveUserKind 값 : 0(일반결재),1(보안결재),2(PCURL결재)  \* Step 3: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 승인과 파일전송 결재에 대한  노티가 확인되면 사용자(결재요청자)가 인식할 수 있도록 화면 오른쪽 하단에  승인완료 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 정보보안 결재 승인완료(SEAC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.SEAC** | **기능번호** | **ET.AN.9** |
| **기능설명** | 개인정보 검출된 결재건에 대해 보안결재자가 승인을 완료하면 서버로부터 승인 완료 노티를 수신받아 해당 사용자(요청자)에게 트레이로 표출한다. | | |
| **입력** | 보안결재 승인완료 노티 | | |
| **출력** | 보안결재 승인완료 알림트레이 | | |
| \* Step 1: 사용자가 DLP정보가 포함된 파일전송을 수행 한 후 보안결재자가 결재(승인)를 수행하면  서버로부터 APPROVEACTION\_NOTIFY 노티를 수신 받는다.  \* Step 2: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 데이터를 파싱하고  ACTION 값을 확인하여 결재종류를 확인한 다음 APPROVEUSERKIND 값을 확인하여  파일전송 보안결재인지 여부도 확인한다.  Cf) Action 값 : 1(승인), 2(반려),3(등록완료),4(등록취소) -> 3,4번은 PCURL 에서만 사용.  ApproveUserKind 값 : 0(일반결재),1(보안결재),2(PCURL결재)  \* Step 3: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 승인과 파일전송 보안 결재에 대한  노티가 확인되면 사용자(결재요청자)가 인식할 수 있도록 화면 오른쪽 하단에  보안결재 승인완료 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | PCURL 사용신청 승인완료(PUAC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.PUAC** | **기능번호** | **ET.AN.10** |
| **기능설명** | PCURL 사용신청 결재건에 대해 서버로부터 승인 완료 노티를 수신받아 해당 사용자(사용신청자)에게 트레이로 표출한다. | | |
| **입력** | PCURL 사용신청 승인완료 노티 | | |
| **출력** | PCURL 사용신청 승인완료 알림트레이 | | |
| \* Step 1: 사용자가 URL 사용 신청을 수행 한 후 결재자가 결재(승인)를 수행하면  서버로부터 APPROVEACTION\_NOTIFY 노티를 수신 받는다.  \* Step 2: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 데이터를 파싱하고  ACTION 값을 확인하여 결재종류를 확인한 다음 APPROVEUSERKIND 값을 확인하여  URL 사용신청 결재 여부도 확인한다.  Cf) Action 값 : 1(승인), 2(반려),3(등록완료),4(등록취소) -> 3,4번은 PCURL 에서만 사용.  ApproveUserKind 값 : 0(일반결재),1(보안결재),2(PCURL결재)  \* Step 3: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 승인과 URL 사용 신청에 대한  노티가 확인되면 사용자(결재요청자)가 인식할 수 있도록 화면 오른쪽 하단에  URL 사용신청 승인완료 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 신규URL 사용신청 등록완료(PURC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.PURC** | **기능번호** | **ET.AN.11** |
| **기능설명** | 신규 URL 사용신청 등록건에 대해 관리자가 해당 신규 URL 확인 후 등록을 수행하면서버로부터 등록완료 노티를 수신받아 해당 사용자(신규 URL 사용신청자)에게 트레이로 표출한다. | | |
| **입력** | 신규 URL 사용신청 등록완료 노티 | | |
| **출력** | 신규 URL 사용신청 등록완료 트레이 | | |
| \* Step 1: 사용자가 신규 URL 사용 신청을 수행 한 후 결재자가 결재(승인)를 수행하면  관리자는 웹매니저상에서 신규URL 정보를 확인한다.  \* Step 2: 사용 신청된 신규 URL 정보를 확인한 관리자는 해당 신규 URL을 등록한다.  \* Step 3: 관리자에 의해 신규 URL 등록이 완료되면 서버로부터 APPROVEACTION\_NOTIFY  노티를 수신 받는다.  \* Step 4: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 데이터를 파싱하고  ACTION 값을 확인하여 결재종류를 확인한 다음 APPROVEUSERKIND 값을 확인하여  신규 URL 사용신청 등록 여부도 확인한다.  Cf) Action 값 : 1(승인), 2(반려),3(등록완료),4(등록취소) -> 3,4번은 PCURL 에서만 사용.  ApproveUserKind 값 : 0(일반결재),1(보안결재),2(PCURL결재)  \* Step 3: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 신규 URL 사용신청 등록에 대한  노티가 확인되면 사용자(결재요청자)가 인식할 수 있도록 화면 오른쪽 하단에  신규 URL 사용신청 등록완료 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 파일전송 결재 반려(FTAR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.FTAR** | **기능번호** | **ET.AN.12** |
| **기능설명** | 파일 전송 건에 대해 서버로부터 반려 노티를 수신받아 해당 사용자(요청자)에게 트레이로 표출한다. | | |
| **입력** | 반려 노티 | | |
| **출력** | 반려 알림 노티 | | |
| \* Step 1: 사용자가 파일전송을 수행 한 후 결재자가 결재(반려)를 수행하면  서버로부터 APPROVEACTION\_NOTIFY 노티를 수신 받는다.  \* Step 2: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 데이터를 파싱하고  ACTION 값을 확인하여 결재종류를 확인한 다음 APPROVEUSERKIND 값을 확인하여  파일전송 결재인지 여부도 확인한다.  Cf) Action 값 : 1(승인), 2(반려),3(등록완료),4(등록취소) -> 3,4번은 PCURL 에서만 사용.  ApproveUserKind 값 : 0(일반결재),1(보안결재),2(PCURL결재)  \* Step 3: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 파일전송 결재에 대한 반려  노티가 확인되면 사용자(결재요청자)가 인식할 수 있도록 화면 오른쪽 하단에  반려되었음을 알리는 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 정보보안 결재 반려(SEAR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.SEAR** | **기능번호** | **ET.AN.13** |
| **기능설명** | 개인정보 검출된 결재건에 대해 보안결재자가 반려하면 서버로부터 반려 노티를 수신받아 해당 사용자(요청자)에게 트레이로 표출한다. | | |
| **입력** | 보안결재 반려 노티 | | |
| **출력** | 보안결재 반려 알림 트레이 | | |
| \* Step 1: 사용자가 DLP정보가 포함된 파일전송을 수행 한 후 보안결재자가 결재(반려)를 수행하면  서버로부터 APPROVEACTION\_NOTIFY 노티를 수신 받는다.  \* Step 2: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 데이터를 파싱하고  ACTION 값을 확인하여 결재종류를 확인한 다음 APPROVEUSERKIND 값을 확인하여  파일전송 보안결재인지 여부도 확인한다.  Cf) Action 값 : 1(승인), 2(반려),3(등록완료),4(등록취소) -> 3,4번은 PCURL 에서만 사용.  ApproveUserKind 값 : 0(일반결재),1(보안결재),2(PCURL결재)  \* Step 3: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 파일전송 보안결재에 대한 반려  노티가 확인되면 사용자(결재요청자)가 인식할 수 있도록 화면 오른쪽 하단에  보안결재 반려 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | PCURL 사용신청 반려(PUAR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.PUAR** | **기능번호** | **ET.AN.14** |
| **기능설명** | PCURL 사용신청 결재건에 대해 서버로부터 반려 노티를 수신받아 해당 사용자(사용신청자)에게 트레이로 표출한다. | | |
| **입력** | PCURL 사용신청 반려 노티 | | |
| **출력** | PCURL 사용신청 반려 알림 트레이 | | |
| \* Step 1: 사용자가 URL 사용 신청을 수행 한 후 결재자가 결재(반려)를 수행하면  서버로부터 APPROVEACTION\_NOTIFY 노티를 수신 받는다.  \* Step 2: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 데이터를 파싱하고  ACTION 값을 확인하여 결재종류를 확인한 다음 APPROVEUSERKIND 값을 확인하여  URL 사용신청 결재 여부도 확인한다.  Cf) Action 값 : 1(승인), 2(반려),3(등록완료),4(등록취소) -> 3,4번은 PCURL 에서만 사용.  ApproveUserKind 값 : 0(일반결재),1(보안결재),2(PCURL결재)  \* Step 3: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 URL 사용 신청에 대한 반려  노티가 확인되면 사용자(결재요청자)가 인식할 수 있도록 화면 오른쪽 하단에  URL 사용신청 반려 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 신규URL 사용신청 등록취소(PURR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.PURR** | **기능번호** | **ET.AN.15** |
| **기능설명** | 신규 URL 사용신청 등록건에 대해 관리자가 해당 신규 URL 확인 후 등록을 취소하면서버로부터 등록취소 노티를 수신받아 해당 사용자(신규 URL 사용신청자)에게 트레이로 표출한다. | | |
| **입력** | 신규 URL 사용신청 등록취소 노티 | | |
| **출력** | 신규 URL 사용신청 등록취소 알림 트레이 | | |
| \* Step 1: 사용자가 신규 URL 사용 신청을 수행 한 후 결재자가 결재(승인)를 수행하면  관리자는 웹매니저상에서 신규URL 정보를 확인한다.  \* Step 2: 사용 신청된 신규 URL 정보를 확인한 관리자는 해당 신규 URL을 등록취소 한다.  \* Step 3: 관리자에 의해 신규 URL 등록이 취소되면 서버로부터 APPROVEACTION\_NOTIFY  노티를 수신 받는다.  \* Step 4: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 데이터를 파싱하고  ACTION 값을 확인하여 결재종류를 확인한 다음 APPROVEUSERKIND 값을 확인하여  신규 URL 사용신청 등록 여부도 확인한다.  Cf) Action 값 : 1(승인), 2(반려),3(등록완료),4(등록취소) -> 3,4번은 PCURL 에서만 사용.  ApproveUserKind 값 : 0(일반결재),1(보안결재),2(PCURL결재)  \* Step 3: APPROVEACTION\_NOTIFY 노티를 수신받은 Agent 는 신규 URL 사용신청 등록취소에 대한  노티가 확인되면 사용자(결재요청자)가 인식할 수 있도록 화면 오른쪽 하단에  신규 URL 사용신청 등록취소 알림 트레이를 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 수동다운로드 파일 수신대기(FTRE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.FTRE** | **기능번호** | **ET.AN.16** |
| **기능설명** | 수동다운로드 사용 일 경우 수신받을 파일이 존재함을 알리는 트레이를 표출한다. | | |
| **입력** | 수동다운로드 파일수신 알림노티 | | |
| **출력** | 수동다운로드 파일수신 알림 트레이 | | |
| \* Step 1: 수동다운로드를 사용 중일 경우 수신 받을 파일이 존재 하면 서버로부터 PUSHFILEINFO  노티를 수신 받는다.  \* Step 2: PUSHFILEINFO 노티를 수신받은 Agent 는 사용자에게 수신받을 파일이 존재함을 알리는  트레이를 화면 오른쪽 하단에 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 수신자 파일 수신대기(FORE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.FORE** | **기능번호** | **ET.AN.17** |
| **기능설명** | 사용자가 파일포워딩 수신자인 경우 수신받을 파일이 존재함을 알리는 트레이를 표출한다. | | |
| **입력** | 수신자 파일수신 알림노티 | | |
| **출력** | 수신자 파일수신 알림 트레이 | | |
| \* Step 1: 사용자가 파일포워딩 수신자일 경우 수신 받을 파일이 존재 하면 서버로부터  FORWARDFILEINFO 노티를 수신 받는다.  \* Step 2: FORWARDFILEINFO 노티를 수신받은 Agent 는 데이터를 파싱하고  FORWARDFILEINFO 노티의 FORWARDCOUNT 값을 확인한다.  \* Step 3: FORWARDCOUNT 값이 확인되면 수신받을 파일건수가 몇건인지 여부를 알려주는  트레이를 화면 오른쪽 하단에 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | APT 검사(APT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.APT** | **기능번호** | **ET.AN.18** |
| **기능설명** | APT 검사 후 악성코드 검출 시 검출내용 메시지를 사용자에게 팝업한다. | | |
| **입력** | 악성코드 검출 알림노티 | | |
| **출력** | 악성코드 검출 알림 팝업 | | |
| \* Step 1: 사용자가 파일 전송 후 파일 전송건에 대해 악성코드 검출 시 서버로부터 APT\_SCAN  노티를 수신받는다.  \* Step 2: APT\_SCAN 노티를 수신받은 Agent 는 수신된 노티 데이터를 파싱하고  검출내용을 알려주는 팝업창을 표출한다.  Cf) 팝업창 표출 내용 – 제목, 검출메시지, 악성코드 정보, 전송취소 알림 메시지 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 바이러스 검사(VIRUS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.VIRUS** | **기능번호** | **ET.AN.19** |
| **기능설명** | 바이러스 검사 후 바이러스 검출 시 검출내용 메시지를 사용자에게 팝업한다. | | |
| **입력** | 바이러스 검출 알림노티 | | |
| **출력** | 바이러스 검출 알림팝업 | | |
| \* Step 1: 사용자가 파일 전송 후 파일 전송건에 대해 바이러스 검출 시 서버로부터 VIRUS\_SCAN  노티를 수신받는다.  \* Step 2: VIRUS\_SCAN 노티를 수신받은 Agent 는 수신된 노티 데이터를 파싱하고  검출내용을 알려주는 팝업창을 표출한다.  Cf) 팝업창 표출 내용 – 제목, 검출메시지, 바이러스 정보, 전송취소 알림 메시지 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 로그인전 결재알림(BLOAN) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.BLOAN** | **기능번호** | **ET.AN.20** |
| **기능설명** | Agent 가 로그인되지 않은 상태에서 결재자에게 결재가 필요한 알림창을 팝업한다. | | |
| **입력** | 사용자 ID (결재건수를 조회하려는 결재자 ID) | | |
| **출력** | 결재건수 알림 트레이 | | |
| \* Step 1: 결재자에게 결재알림창을 표현하는 SGApproveNotify 실행 파일을 시작프로그램에 등록한다.  \* Step 2: 결재자가 PC를 부팅하여 시작할 때 SGApproveNotify 팝업 창이 실행되면서  개인설정정보가 저장된 SecureGate.ini 파일에서 최근에 로그인한 사용자 ID를 로드한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK  \* Step 3: 로드된 사용자 ID 정보로 weblink를 통해 해당 사용자의 승인대기 건수를 조회한다.  Cf) 조회 요청 URL : <https://IP>주소/weblink/CM001\_approveCount.do?id=사용자ID  \* Step 4: Step3 에서 조회된 응답 값을 파싱하여 결재건수,이름,직위 정보를 로드한다.  Cf) 조회 요청 응답.  - resultCode : 0(성공), 1(사용자ID 없음), 2(사용자 기한만료), 3(시스템 오류)  - approveCount = 1 (승인대기 건수)  - UserName = 홍길동 (결재자 이름)  - UserRank = 과장 (결재자 직위)  \* Step 5: 응답값이 정상(resultCode)이고 승인대기 건수(approveCount)가 하나 이상이면  승인대기 알림 트레이를 화면 오른쪽 하단에 표출한다.  \* Step 6: 사용자가 알림팝업을 클릭하면 Agent를 실행하여 로그인창을 표출한다.  이미 Agent 가 실행 중이면 Agent 를 전면으로 띄운다.  단, 결재알림 팝업에서 로드된 ID 가 현재 로그인 되어 실행중인 Agent ID 와 다를 경우  사용자에게 ID 가 다름을 결재알림 팝업에 표출한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 알림 노티(AN) | 화면이동(MOVE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.AN.MOVE** | **기능번호** | **ET.AN.21** |
| **기능설명** | 각각의 알림 노티에 따라 트레이 클릭 시 자동으로 화면을 전환한다.  Ex) 승인대기 트레이 클릭 -> 결재관리 화면으로 이동. | | |
| **입력** | 알림 트레이 클릭. | | |
| **출력** | Agent 화면 자동이동. | | |
| \* Step 1: 서버로부터 알림 노티를 수신받는다.  Cf) 트레이로 표출되는 서버 노티  - APPROVECOUNT : 파일전송 승인대기 알림 노티, PCURL 사용신청 승인대기 알림 노티.  - EMAIL\_APPROVE\_NOTIFY : 이메일 승인대기 알림 노티.  - PRIVACY\_APPROVE\_NOTIFY : 보안결재 승인대기 알림 노티.  - APPROVEACTION\_NOTIFY : 일반파일전송 결재, 보안결재, PCURL 결재 완료 노티.  - PUSHFILEINFO : 수신 대기 알림(수동다운로드 사용시)  - FORWARDFILEINFO : 수신자 수신대기 알림.(파일 포워등 사용 시)  \* Step 2: Agent 는 서버로부터 노티를 수신받고 적절한 알림 트레이창을 화면 오른쪽 하단에  표출한다.  \* Step 3: 표출된 트레이를 클릭하면 Agent는 각각의 알림트레이에 맞는 화면으로 자동이동한다.  Cf) 이동 화면   * 파일전송 승인대기 알림 트레이 클릭 -> 파일 결재관리 화면 이동. * 이메일 승인대기 알림 트레이 클릭 -> 이메일 결재관리 화면 이동 * 보안결재 승인대기 알림 트레이 클릭 -> 보안 결재관리 화면 이동. * PCURL 사용신청 승인대기 알림 트레이 클릭 -> PCURL 결재관리 화면 이동. * 파일전송 결재완료 알림 트레이 클릭 -> 파일 전송대기 화면 이동. * 보안결재 결재완료 알림 트레이 클릭 -> 파일 전송대기 화면 이동. * PCURL 결재완료 알림 트레이 클릭. -> PCURL 신청관리 화면 이동. * 수신대기 알림 트레이 클릭 -> 파일 전송대기 화면 이동. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | URL List 실시간 갱신(UREU) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.UREU** | **기능번호** | **ET.CF.1** |
| **기능설명** | URL List 체크항목을 주기적으로 갱신 및 적용한다. | | |
| **입력** | URL.xml 갱신 시간 | | |
| **출력** | 갱신된 URL List 수신. | | |
| \* Step 1: Agent 는 실행 중 LinkCheck 를 통해 URL.xml(URL List) 파일을 갱신한다.  \* Step 2: Agent에서 세션 유지를 위해 LinkChk 를 요청하고  LinkChk 응답으로 UrlTimeStamp 값을 수신 받는다.  Cf) UrlTimeStamp 값은 서버에서 URL List 갱신 시 변경된다.  \* Step 3: Agent는 로컬에 저장되어 있는 URL.xml 파일의 마지막 갱신 시간을 로드하여  LinkChk 응답 후 수신받은 UrlTimeStamp 시간 값과 비교한다.  \* Step 4: Agent는 LinkChk 응답 후 수신받은 UrlTimeStamp 시간 값이 더 최신일 경우  프로토콜 “URLLIST”를 사용하여 새로 URL List 목록을 갱신한다.  \* Step 5: URL List 목록이 새로 갱신되면 URL List 마지막 갱신 시간을 변경한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 파일 위변조 체크항목 실시간 갱신(FREU) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.FREU** | **기능번호** | **ET.CF.2** |
| **기능설명** | 파일 위변조 체크항목을 주기적으로 갱신 및 적용한다. | | |
| **입력** |  | | |
| **출력** |  | | |
| # 프로그램의 내부 구현을 기술한다.  자료구조와 관계형 데이터베이스 모델과 프로그래밍 언어, 도구, 알고리즘 선택과 같은 항목을 다룬다  \* 입력: SRV IP( VIP), SRV Port  \* 출력: Real SRV IP  \* Step 1: 입력 되어진 서버IP와 포트에 대해 입력값 벨리데이션을 한다.  \* Step 2: 소켓을 생성하여 망연계 VIP로 접속하여(SSL), 1002(접속IP 정보 가져오기) 메시지를 전송한다.  \* Step 3: 응답(2002)를 대기하고, 2002수신후 Real IP를 리턴한다.  \* Step 4: 2002응답이 Timeout시간동안 기다리고 응답이 수신되지않으면, 에러코드: 미응답으로 리턴한다  위의 내용을 그림으로 표현해도 좋음. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 다국어 지원(LANG) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.LANG** | **기능번호** | **ET.CF.3** |
| **기능설명** | Agent 는 다국어를 지원한다.(영어/일본어/중국어) | | |
| **입력** | 언어 설정정보 | | |
| **출력** | 설정정보에 따라 UI 언어 출력. | | |
| \* Step 1: Agent 시작 시 설정된 언어 정보를 로드하여 해당 설정에 맞는 언어 정보를 로딩하여  UI 에 표출한다.  Cf) 언어 설정 정보는 개인설정정보가 저장된 SecureGate.ini 의 LANGUAGE 값을 참조한다.  \* Step 2: SecureGate.ini 에 LANGUAGE 값이 설정되어 있지 않은 경우에는  Agent 시작 시 SGConf.xml 의 설정 정보를 SecureGate.ini 에 저장하고  언어 설정정보를 로드한다.  \* Step 3: SecureGate.ini 에 설정된 LANGUAGE 값에 따라 HSText.xml 에 저장된 언어를 로드한다.  Cf) 현재 지원되는 언어 – 영어,일본어, 중국어 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | APT 검사 진행(APT) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.APT** | **기능번호** | **ET.CF.4** |
| **기능설명** | APT 검사 진행 상황을 전송관리 조회 시 사용자에게 표출한다. | | |
| **입력** | 전송관리 조회 요청 | | |
| **출력** | APT 검사 진행 상태정보 | | |
| \* Step 1: 전송관리 조회 시 APT 검사 진행상황 확인이 가능하다.  \* Step 2: 전송관리 조회 요청 후 Agent 는 서버 DB 쿼리 조회를 통해 APT 진행상황의 기준치 값을  가져온다.  \* Step 3: 서버로부터 기준치 조회가 완료되면 DB 쿼리 조회를 통해 임시폴더에 APT 검사 진행중인  파일개수를 확인한다.  \* Step 4: Step2 에서 얻은 기준값과 Step3 에서 얻은 파일개수를 비교하여 상태값을 구한다.  Cf) 상태값 : 0(조회 실패), 1(원할), 2(지체), 3(혼잡)  \* Step 5: Step4에서 얻어진 상태값을 전송관리 조회 항목에 표출한다.  Cf) APT 상태 기준값 조회 쿼리   |  | | --- | | CREATE TABLE tbl\_apt\_status\_limit  ( nlevel integer, nfrom integer,nto integer)WITH (OIDS=FALSE);  ALTER TABLE tbl\_apt\_status\_limit  OWNER TO hsck; |   Cf) APT 상태값 조회 쿼리   |  | | --- | | CREATE TABLE tbl\_apt\_status\_mds  ( mdsnum integer, vmqval integer) WITH ( OIDS=FALSE );  ALTER TABLE tbl\_apt\_status\_mds  OWNER TO hsck; | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 파일과 클립보드 일일 전송 가능한 용량 및 횟수 표시(DFCV) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.DFCV** | **기능번호** | **ET.CF.5** |
| **기능설명** | Agent는 Status bar 에 일일 전송 가능한 파일과 클립보드 전송량 및 횟수를 표시한다. | | |
| **입력** | 파일과 클립보드 전송량 및 횟수 | | |
| **출력** | 갱신된 파일과 클립보드 전송량 및 횟수 | | |
| \* Step 1: Agent 는 파일 전송 또는 클립보드 전송 시 일일 전송 가능 사용량 및  일일 전송 가능 횟수를 확인한다.  \* Step 2: 파일과 클립보드 전송에 대한 일일 전송 가능 사용량 및 일일 전송 가능 횟수는  Agent 로그인 시 개인화된 사용자 설정정보를 통해 서버로부터 수신받는다.  Cf) bind\_ack 시 일일 전송량 및 일일 전송횟수 태그값  - DAYFILETRANSFERLIMITSIZE : 일일 파일 전송가능 사이즈 ( 0: 무제한,1이상-단위MB )  - DAYFILETRANSFERLIMITCOUNT : 일일 파일전송 가능한 횟수 ( 0 : 무제한, 1이상 )  - DAYCLIPBOARDSIZE : 일일 클립보드 전송가능 사이즈 ( 0: 무제한,1이상-단위MB )  - DAYCLIPBOARDCOUNT : 일일 클립보드 전송 가능한 횟수 ( 0 : 무제한, 1이상 )  \* Step 3: Agent 는 파일 전송 또는 클립보드 전송 시 서버 DB 조회를 통해 파일과 클립보드 전송에  관한 사용량 및 횟수의 유효성을 검사한다.  \* Step 4: 유효성 검사가 완료되면 Agent 는 전송 후 서버 DB 조회를 통해 Status bar 에 표시된  파일과 클립보드 전송량 및 횟수를 갱신한다.  Cf) 일일 가능 파일 전송 사용량 및 횟수 조회 쿼리   |  | | --- | | "SELECT U.USER\_ID, SUM(F.FILE\_SIZE) AS FS, COUNT(\*) AS CNT \n\  FROM ( \n\  SELECT 'H' AS TPOS, TRANS\_SEQ, REQUEST\_TIME, USER\_SEQ, TRANS\_FLAG, RECV\_FLAG, PCTRANS\_FLAG,APPROVE\_FLAG, SYSTEM\_ID \n\  FROM TBL\_TRANSFER\_REQ\_HIS H WHERE TRANS\_SEQ BETWEEN '##DATE##0000000000' AND '##DATE##9999999999' \n\  UNION ALL \n\  SELECT 'C' AS TPOS, TRANS\_SEQ, REQUEST\_TIME, USER\_SEQ, TRANS\_FLAG, RECV\_FLAG, PCTRANS\_FLAG,APPROVE\_FLAG, SYSTEM\_ID \n\  FROM TBL\_TRANSFER\_REQ\_INFO T WHERE TRANS\_SEQ BETWEEN '##DATE##0000000000' AND '##DATE##9999999999' \n\  ) T \n\  , TBL\_USER\_INFO U \n\  , ( \n\  SELECT TRANS\_SEQ, SUM(F.FILE\_SIZE) AS FILE\_SIZE \n\  FROM TBL\_FILE\_LIST\_HIS F WHERE FILE\_SEQ BETWEEN '##DATE##0000000000' AND '##DATE##9999999999' \n\  GROUP BY F.TRANS\_SEQ \n\  ) F \n\  WHERE T.TRANS\_SEQ=F.TRANS\_SEQ \n\  AND T.REQUEST\_TIME BETWEEN '##DATE##0000' AND '##DATE##235959' \n\  AND U.USER\_SEQ=T.USER\_SEQ \n\  AND U.USER\_SEQ='##USERSEQ##' \n\  AND FUNC\_TRANSSTATUS(T.TRANS\_FLAG, T.RECV\_FLAG, T.PCTRANS\_FLAG) NOT IN ('C', 'F') \n\  AND T.APPROVE\_FLAG !='3' \n\  AND SUBSTRING(T.SYSTEM\_ID, 1, 2)='##SYSID##' \n\  GROUP BY U.USER\_ID " |   Cf) 일일 가능 클립보드 전송 사용량 및 횟수 조회 쿼리.   |  | | --- | | "SELECT USER\_ID, SUM(DATA\_SIZE) AS DATA\_SIZE, COUNT(\*) AS CNT \n\  FROM TBL\_CLIPBOARD\_HIS \n\  WHERE WORK\_ID BETWEEN '##DATE##0000000000' AND '##DATE##9999999999' \n\  AND SUBSTRING(SYSTEM\_ID, 1,2)='##SYSID##' \n\  AND DATA\_TYPE IN ('1', '2') \n\  AND USER\_ID = (SELECT USER\_ID FROM TBL\_USER\_INFO WHERE USER\_SEQ='##USERSEQ##')\n\  GROUP BY USER\_ID" | | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 트레이 아이콘 전환(TC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.TM** | **기능번호** | **ET.CF.6** |
| **기능설명** | 사용자가 원하는 설정에 따라 시작 또는 종료 시 트레이 아이콘으로 전환한다. | | |
| **입력** | 설정 정보 | | |
| **출력** | 트레이 아이콘 전환. | | |
| \* Step 1: Agent는 설정된 정보를 통해 로그인 시 또는 종료 버튼 클릭 시 트레이 아이콘 전환기능이  가능하다.  \* Step 2: Agent는 SGConf.xml 의 START TRAY 사용 유무를 확인하고 로그인 시 트레이 아이콘 상태로  자동 전환된다.  \* Step 3: 종료 버튼 클릭 시 트레이 아이콘 전환 기능은 환경설정 정보를 통해 변경이 가능하며  개인설정정보가 저장된 SecureGate.ini 파일의 태그 USETRAY 값을 확인하여  저장된 설정정보에 맞게 동작한다.  Cf) USETRAY 값 : Y – 종료 버튼 클릭 시 트레이 아이콘 전환.  N – 종료 버튼 클릭 시 Agent 종료. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | Agent 재시작(AAE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.AAE** | **기능번호** | **ET.CF.7** |
| **기능설명** | Agent 실행 상태를 모니터링하여 실행 오류 발생 시 Agent 를 자동으로 재시작한다. | | |
| **입력** |  | | |
| **출력** |  | | |
| \* Step 1: Agent 설치 시 Agent 실행 상태를 모니터링하는 HSMonSvc 서비스 실행 파일을  window 서비스로 등록하고 자동 시작되도록 설정한다.  \* Step 2: Agent 는 시작 시부터 완료 시점 까지 SecureGateStatus.ini 파일에  STATUS 태그값으로 각 단계의 상태값을 저장한다.  Cf) STATUS 태그값 : 0(시작), 1(로그인 대기), 2(로그인 실패 종료), 3(UI 생성중), 4(완료),  5(업데이트 중)  SecureGateStatus.ini 경로 : Agent 설치경로(C:\HANSSAK\SecureGate)  \* Step 3: Window 서비스로 등록된 HSMonSvc 는 SecureGateStatus.ini 파일의 STATUS 값을  3분 간격으로 로드한다.  \* Step 4: HsMonSvc 서비스는 상태가 로그인대기(1), 완료(4), 업데이트 중(5) 을 제외하고  현재 로드한 값이 시작(0) 또는 로그인 실패 종료(2), UI생성중(3) 일 경우  3분전에 로드한 상태 값과 비교하여 동일할 경우 Agent 를 재실행한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 파일 수신경로 변경(RFC) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.RFC** | **기능번호** | **ET.CF.8** |
| **기능설명** | 환경설정의 기능을 통해 사용자가 원하는 경로로 파일 수신 경로를 변경한다. | | |
| **입력** | 사용자 변경. | | |
| **출력** | 파일 수신경로 변경. | | |
| \* Step 1: 환경설정 팝업 창에서 수신위치가 표현된 에디트 옆의 버튼을 클릭하면  파일 수신위치 변경을 위한 폴더탐색기 창이 팝업된다.  \* Step 2: 팝업 된 폴더탐색기 창에서 파일 수신경로를 변경하면 수신위치가 표현된 에디트의  수신위치가 변경된다.  \* Step 3: 사용자가 Agent 환경설정창의 확인 버튼을 누르면 변경된 수신위치가 DownPath.ini 파일에  UTF-8 형태로 저장된다.  Cf) DownPath.ini 경로 : %USERPROFIEL%\AppData\LocalLow\HANSSAK  \* Step 4: 변경된 파일 수신위치가 DownPath.ini 파일에 저장되면 Agent 는 메모리상의 파일 수신위치  경로를 갱신한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 파일 수신 후 폴더 열기(RFCO) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.RFCO** | **기능번호** | **ET.CF.9** |
| **기능설명** | 파일 수신이 완료되면 자동으로 파일 수신 경로의 폴더를 Open 한다.  환경설정을 통해 사용 유무 설정이 가능하다. | | |
| **입력** | 사용자 변경. | | |
| **출력** | 파일 수신 후 탐색기 폴더 자동 Open. | | |
| \* Step 1: 환경설정 팝업창에서 수신위치가 표현된 에디트 아래의 “파일 수신 후 폴더 열기”  체크박스를 통해 파일 수신 후 탐색기 폴더 자동 Open 기능이 가능하다.  \* Step 2: 사용자에 의해 “파일 수신 후 폴더 열기” 체크 및 해제후 환경설정창의 확인 버튼을  누르면 개인설정정보가 저장된 SecureGate.ini 파일에 DOWNOPEN 태그 값에 저장된다.  Cf) DOWNOPEN 태그값  - 체크 ON 상태 일 때 Y, 체크 OFF 상태일 때 N | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 승인대기 알림 설정(SNSE) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.SNSE** | **기능번호** | **ET.CF.10** |
| **기능설명** | 환경설정의 기능을 통해 사용자가 승인대기 알림 사용 유무를 설정한다. | | |
| **입력** | 사용자 변경. | | |
| **출력** | 승인대기 알림 노티 사용 유무 | | |
| \* Step 1: 환경설정 팝업창에서 “승인대기 알림사용” 체크박스를 통해 파일 수신 후  탐색기 폴더 자동 Open 기능이 가능하다.  \* Step 2: 사용자에 의해 “승인대기 알림사용” 체크 및 해제후 환경설정창의 확인 버튼을  누르면 개인설정정보가 저장된 SecureGate.ini 파일에 APPROVENOTIVIEW 태그 값에 저장된다.  Cf) APPROVENOTIVIEW 태그값  - 체크 ON 상태 일 때 1, 체크 OFF 상태일 때 0 | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 접속환경(CENV) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.CENV** | **기능번호** | **ET.CF.11** |
| **기능설명** | Agent 버전정보 등 설정된 전체 또는 개인별 설정 정보를 보여준다. | | |
| **입력** | 메뉴 실행 | | |
| **출력** | 접속환경 팝업창. | | |
| \* Step 1: Agent 실행 중 도구 메뉴의 “접속환경” 메뉴를 통해 버전정보 또는 개인별  설정정보 확인이 가능하다.  \* Step 2: “도구” -> “접속환경” 메뉴를 실행하면 접속환경 정보 팝업창이 표출된다.  Cf) 접속환경 창에서 표출되는 정보   * 버전 정보 : 현재 실행중인 Agent 버전 정보를 나타냄 * 접속 서버 : 접속된 서버가 내부인지 외부인지 정보를 표출. * 전송최대용량 : 한 번에 전송가능한 최대용량 표시. * 파일개수제한 : 한 번에 전송가능한 파일 최대 개수 표시. * 결재관리 : 결재 사용유무 표시. * 다운로드 파일관리 : 수신파일 삭제 유무 및 삭제 주기 표시. * 클립보드 : 클립보드 사용 유무 표시. * 클립보드 최대용량: 한 번에 전송가능한 클립보드 최대 용량 표시. * URL 자동전환 : URL 자동전환 사용유무 표시. * 바이러스 검사 : 바이러스 검사 유무 표시. * 일일 파일 전송횟수 제한 : 일일 가능한 파일 전송 횟수 표시. * 일일 파일 전송용량 제한 : 일일 가능한 파일 전송 용량 표시. * 일일 클립보드 전송횟수 제한 : 일일 가능한 클립보드 전송 횟수 표시. * 일일 클립보드 전송용량 제한 : 일일 가능한 클립보드 전송 용량 표시. * 확장자 제한 : 확장자제한 타입(Black/White) 및 확장자 제한 종류 표시. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 파일마임정보 수신(FMR) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.FMR** | **기능번호** | **ET.CF.12** |
| **기능설명** | 서버에서 추가 설정한 파일마임정보를 수신받아 파일마임정보를 추가한다. | | |
| **입력** | 추가 파일마임정보 요청. | | |
| **출력** | 추가된 파일마임정보 갱신. | | |
| \* Step 1: Agent 는 로그인 수행 전 서버에 추가된 파일마임정보를 요청한다.  \* Step 2: 3436 포트를 통해 망연계 VIP 로 접속하여 1103 메시지를 전송하여 추가 파일마임정보를  요청한다.  \* Step 3: Agent 는 2103 응답을 수신 받고 수신받은 정보를 설치디렉토리 Conf 폴더에  FileMime.conf 에 저장한다.  \* Step 4: FileMime.conf 파일 생성이 완료되면 Agent 는 해당 파일의 마임정보를 메모리상에 추가한다. | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | 파일마임정보 수신(USS) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.USS** | **기능번호** | **ET.CF.13** |
| **기능설명** | 환경설정에서 설정한 정보들을 사용자별로 별도파일에 저장한다. | | |
| **입력** | Agent 설정 정보 | | |
| **출력** | Agent 설정 정보 저장. | | |
| \* Step 1: Agent 는 사용자 환경설정 정보 수정에 따라 별도의 사용자별 설정정보를 저장한다.  \* Step 2: 환경설정 수정을 통해 변경된 저장정보는 SecureGate.ini 파일에 저장한다.  \* Step 3: SecureGate.ini 파일은 %USERPROFILE%\AppData\LocalLow\HANSSAK 경로에 생성된다.  Cf ) SecureGate.ini 에 저장되는 설정값   * LANGUAGE : 언어 설정정보 – KR(한국어), JP(일본어), EN(영어), CN(중국어) * AGENTHWNDD : 현재 실행중인 Agent Window Handle 값 * LOCATIION : Agent 설치파일 위치정보 ( 내부 : IN, 외부 : EX ) * DOWNOPEN : 파일 수신 후 폴더열기 사용유무 ( 사용 : Y, 사용안함 : N ) * URLAUTOUSE : URL 리다이렉션 사용 유무 ( 사용 : Y, 사용안함 : N ) * SCRLOCKUSE : 화면잠금 사용유무 ( 사용 : Y, 사용안함 : N ) * SCRLOCKVAL : 화면잠금 사용 시 사용될 시간 ( 단위 분 ) * USETRAY : 종료 버튼 클릭 시 Agent 트레이 아이콘 전환 유무 ( 전환 : Y, 사용안함 : N ) * CLIPBOARD : 클립보드 단축키 설정 정보. * ACCESSPATH : 사용자가 자주 사용하는 즐겨찾기 경로 표시. * URLTYPE : URL 리다이렉션 수행 후 익스플로러 처리 타입.(C:닫기, F: 특정페이지로 포워딩, N : 그대로 현상 유지). * UID : ID 저장 또는 자동로그인 시 사용되는 사용자 ID. 암호화 되어 저장. * UPW : 자동로그인 시 사용되는 사용자 PW. 암호화 되어 저장. * AUTOLOGINFLAG : 자동로그인 실행 상태값 ( 0 : 로그인 창 표출, 1: 자동로그인 수행 ) | | | |

|  |  |  |
| --- | --- | --- |
| **대분류** | **중분류** | **소분류** |
| ETC(ET) | 편의기능(CF) | Login 창에 사용자 ID 자동 표출.(IDAU) |

|  |  |  |  |
| --- | --- | --- | --- |
| **기능코드** | **ET.CF.IDAU** | **기능번호** | **ET.CF.14** |
| **기능설명** | id 히스토리 저장 사용시 마지막에 사용한 사용자의 ID를 LOGIN 창에 표출. | | |
| **입력** | 사용자 ID | | |
| **출력** | Login 창에 사용자 ID 자동 표출. | | |
| \* Step 1: Agent 는 사용자 ID 히스토리 저장 기능 사용 시 현재 사용자 ID 를  개인설정 정보가 저장된 SecureGate.ini 파일에 UID 태그값으로 암호화하여 저장한다.  Cf) SecureGate.ini 경로 : %USERPROFILE%\AppData\LocalLow\HANSSAK  \* Step 2: Agent 종료 후 재시작을 하게되면 SecureGate.ini 값의 UID 값을 로드한다.  \* Step 3: 로드한 UID 값 복호화가 완료되면 Agent 로그인 창 ID 입력 에디트 창에  자동으로 복호화된 사용자 ID 를 표출한다. | | | |